



①9 BUNDESREPUBLIK
DEUTSCHLAND



DEUTSCHES
PATENT- UND
MARKENAMT

⑫ **Offenlegungsschrift**
⑩ **DE 198 36 882 A 1**

⑤1 Int. Cl. 7:
H 02 P 7/00

②1 Aktenzeichen: 198 36 882.8
②2 Anmeldetag: 14. 8. 1998
④3 Offenlegungstag: 17. 2. 2000

DE 198 36 882 A 1

⑦1 Anmelder:
Papst-Motoren GmbH & Co. KG, 78112 St Georgen,
DE

⑦4 Vertreter:
Raible, H., Dipl.-Ing., Pat.-Anw., 70192 Stuttgart

⑦2 Erfinder:
Hornberger, Jörg, Dipl.-Ing., 72280 Dornstetten, DE;
Jeske, Frank, Dipl.-Ing., 78112 St Georgen, DE;
Kaltenbrunner, Hansjörg, 78052
Villingen-Schwenningen, DE; Karwath, Arno,
Dipl.-Ing., 78628 Rottweil, DE; Rappenecker,
Hermann, Dipl.-Ing., 78147 Vöhrenbach, DE

⑤6 Für die Beurteilung der Patentfähigkeit in Betracht
zu ziehende Druckschriften:

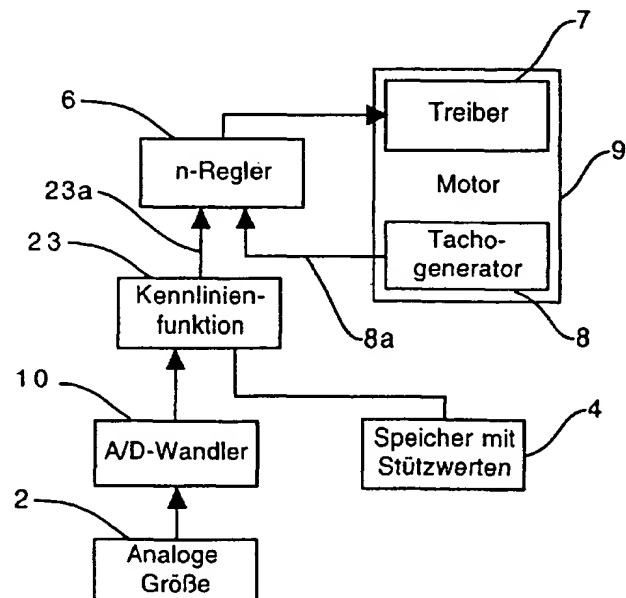
DE 30 03 102 C2
DE 196 47 983 A1
DE 196 00 243 A1
DE 44 41 372 A1
DE 40 25 263 A1

CLAUSSEN, U., FROMME, G.: Motorregelung mit
Mikro-
rechner. In: Regelungstechnische Praxis, 1978,
H. 12, S. 355-359;

Die folgenden Angaben sind den vom Anmelder eingereichten Unterlagen entnommen

⑤4 Anordnung mit einem Elektromotor

⑤7 Die Drehzahl eines Elektromotors (9) wird durch einen Regler (6) geregelt, der seinen Sollwert von einer Kennlinienfunktion (23) erhält. Die Kennlinienfunktion (23) berechnet, ausgehend von einer durch einen A/D-Wandler (10) digital gewandelten, ursprünglich analogen Größe (2), einen Sollwert für den Regler (6) mit Hilfe von in einem Speicher (4) gespeicherten Stützwerten einer Kennlinie, wobei die nicht durch die Stützwerte vorgegebenen Werte durch Interpolation berechnet werden.



DE 198 36 882 A 1

Beschreibung

Die Erfindung betrifft einen Elektromotor und insbesondere einen Elektromotor mit einer Kennlinienfunktion.

Zum Stand der Technik

Beispiele für elektronisch kommutierte Motoren zeigen beispielsweise folgende Schriften der Anmelderin:

DE 44 41 372 A1 (intern: D183)
EP 0 658 973 B1 (intern: EP184)
DE-U 296 06 939.6 (intern: D190i)
DE 195 15 944 A1 (intern: D192)
EP 0 741 449 A1 (intern: EP193)
EP 0 744 807 B1 (intern: EP194)
DE 195 18 991 A1 (intern: D195)
DE 196 47 983 A1 (intern: D199i)
EP 0 780 962 A2 (intern: EP200)

Es wäre nicht möglich, den umfangreichen Inhalt dieser Schriften auch nur in zusammengefaßter Form in der vorliegenden Anmeldung widerzugeben, und deshalb wird auf ihren Inhalt in vollem Umfang Bezug genommen.

Bei der DE 44 41 372 A1 wird das Drehzahlverhalten eines Elektromotors durch eine Kennlinie $n = f(T)$ dargestellt, wobei jeder Temperatur T eine Solldrehzahl n zugeordnet ist. So kann z. B. bei einem Lüfter bei steigender Temperatur die Drehzahl erhöht werden. Das Drehzahl-Temperatur-Verhalten wird hierbei durch analoge Bauelemente bestimmt. Hier ist jedoch durch Fertigungstoleranzen der Bauelemente keine große Genauigkeit erreichbar, und eine Umstellung auf ein anderes Verhalten ist nur mit großem Aufwand möglich.

Es ist deshalb eine Aufgabe der Erfindung, einen neuen Elektromotor, und ein Verfahren zum Betreiben eines solchen Elektromotors, bereitzustellen.

Nach einem ersten Aspekt der Erfindung wird diese Aufgabe gelöst durch einen Elektromotor mit einer von einer variablen physikalischen Größe, insbesondere einer Temperatur, gesteuerten Drehzahl, mit einem in Form von digitalen Einzelwerten gespeicherten Kennlinienfeld zum Zuordnen von Werten dieser physikalischen Größe zu entsprechenden Drehzahlwerten des Elektromotors, und mit einem Mikrocontroller oder Mikroprozessor, der Zugriff zu den gespeicherten digitalen Einzelwerten hat. Ein solcher Elektromotor ist sehr variabel, da durch das gespeicherte Kennlinienfeld sein Drehzahlverhalten als Funktion der variablen physikalischen Größe leicht veränderbar ist. Dabei hat es sich als besonders vorteilhaft erwiesen, die digitalen Einzelwerte mindestens teilweise in Vektorform zu speichern, da dies ihre Verarbeitung wesentlich vereinfacht.

Die Erfindung betrifft ferner ein Verfahren zum Regeln einer physikalischen Größe, insbesondere einer Drehzahl, mit folgenden Schritten:

- a) Aus einem gewünschten Wert für die physikalische Größe in digitaler Form und einem tatsächlichen Wert für die physikalische Größe, ebenfalls in digitaler Form, wird eine Differenz gebildet, um die Regelabweichung zu ermitteln;
- b) Vorzeichen und Absolutwert dieser Differenz werden ermittelt;
- c) abhängig vom Vorzeichen wird ein analoges Speicherglied geladen oder entladen;
- d) die Dauer der Ladung oder Entladung erfolgt jeweils im wesentlichen proportional zur Größe des ermittelten Absolutwerts der Differenz;

e) mit einem von der Ladung des analogen Speicherglieds abhängigen Wert wird das Tastverhältnis eines Stellglieds beeinflusst, welches seinerseits mit seinem Ausgangssignal die zu regelnde physikalische Größe beeinflusst. Hierdurch ergibt sich eine sehr vorteilhafte Kombination zwischen digitaler Genauigkeit bei der Ermittlung der Regelabweichung, und anschließender Verarbeitung dieser Regelabweichung zum Beeinflussen der physikalischen Größe.

Eine weitere Lösung der gestellten Aufgabe ergibt sich durch ein Verfahren zum temperaturabhängigen Steuern der Drehzahl eines Elektromotors, mit folgenden Schritten:

- a) In einem Speicher werden Wertecuster von Kennlinien-Definitionspunkten gespeichert, welche Wertecuster mindestens einen für eine bestimmte Temperatur kennzeichnenden Wert und eine dieser Temperatur zugeordnete Drehzahlinformation enthalten;
- b) ein für die die Motordrehzahl steuernde Temperatur kennzeichnender aktueller Wert wird in zeitlichen Abständen erfaßt;
- c) dieser erfaßte Wert wird mit den gespeicherten, für die Temperatur kennzeichnenden Werten verglichen, welche in den gespeicherten Wertecustern enthalten sind;
- d) ein dem aktuellen Wert benachbarter gespeicherter Wert wird ermittelt;
- e) durch eine von diesem benachbarten Wert ausgehende Interpolation wird eine Drehzahlinformation zu dem erfaßten aktuellen Wert ermittelt;
- f) ein von dieser interpolierten Drehzahlinformation abgeleiteter Wert wird zur Vorgabe einer Solldrehzahl einem Drehzahlregler des Elektromotors zugeführt. Man kann so durch Speicherung einer kleinen Zahl von Wertecustern das Drehzahlverhalten eines Motors abhängig von der Temperatur festlegen.

Ferner betrifft die Erfindung ein Verfahren zur A/D-Wandlung bei einer Anordnung mit einem einen temperaturabhängigen Widerstand enthaltenden Spannungsteiler, wobei ein Abgriff dieses Spannungsteilers das Potential an dem einen Eingang eines Komparators festlegt und das Potential an dem anderen Eingang des Komparators durch einen Kondensator bestimmt wird, der über eine Konstantstromquelle aufladbar ist, mit folgenden Schritten:

- a) Zuerst wird der Kondensator entladen;
- b) anschließend wird die Zeit gemessen, die der Kondensator bei Ladung durch die Konstantstromquelle benötigt, um das Potential des anderen Eingangs zu erreichen;
- c) diese Zeit wird als Maß für die Temperatur des temperaturabhängigen Widerstands verwendet. Ein solches Verfahren kann in einfacher Weise durch einen Mikrocontroller realisiert werden, der Funktionen eines Elektromotors steuert oder regelt. Dieser Mikrocontroller kann im Schritt a) die Entladung des Kondensators bewirken und zur Zeitmessung gemäß Schritt c) dienen, wodurch sich insgesamt ein sehr einfaches Verfahren ergibt.

Eine andere Lösung der gestellten Aufgabe ergibt sich durch einen Elektromotor mit in einem Speicher abgespeicherten diskreten Werten, welche in Form von Stützwerten eine Temperatur/Drehzahl-Kennlinie definieren, wobei die diskreten Werte über eine Datenverbindung zu einem außerhalb des Elektromotors angeordneten Eingabegerät verän-

derbar sind. Dies ermöglicht eine einfache Anpassung eines solchen Motors an unterschiedliche Kundenforderungen.

Eine weitere Lösung der gestellten Aufgabe betrifft ein Verfahren zum Betrieb eines elektronisch kommutierten Motors, welchem ein Mikroprozessor oder Mikrocontroller und ein diesem zugeordnetes Programm zugeordnet sind, welches Programm zur Steuerung einer Mehrzahl von Motorfunktionen unterschiedlicher Priorität dient, mit folgenden Schritten:

- a) Es wird eine Mehrzahl von für den Betrieb des Motors erforderlichen, anforderbaren Routinen vorgesehen;
- b) wenn eine anforderbare Routine benötigt wird, wird ein entsprechendes Anforderungssignal für sie gesetzt;
- c) durch eine übergeordnete Programmfunktion wird geprüft, welche angeforderte Routine die höchste Priorität aufweist, und diese Routine höchster Priorität wird als erste ausgeführt;
- d) nach Ausführung dieser Routine höchster Priorität wird das dieser Routine zugeordnete Anforderungssignal rückgesetzt.

Ein solches Verfahren nützt die vorhandene Rechenkapazität eines Mikroprozessors oder Mikrocontrollers sehr gut aus und ermöglicht es, bestimmte zeitkritische Abfragen oder dergleichen in Abständen zu wiederholen, die eine vorgegebene Zeitdauer nicht überschreiten. Das können z. B. Abfragen eines Datenbusses sein, über den dem Motor Daten oder Befehle zuführbar sind. Dabei wird dieses Verfahren bevorzugt nach Art einer Schleife im Betrieb des Motors ständig wiederholt, wobei sich je nach Art der angeforderten Routine unterschiedliche Abläufe der Schleife ergeben. Besonders vorteilhaft ist es hierbei, wenn eine im Programm auszuführende anforderbare Routine während ihrer Ausführung ihrerseits ein Anforderungssignal für eine andere auszuführende anforderbare Routine erzeugen kann. Dies ermöglicht eine enge Verkettung von Routinen, zwischen denen jeweils zeitkritische Programmschritte ablaufen können.

Eine weitere Lösung der gestellten Aufgabe betrifft einen Motor mit einem Mikroprozessor oder Mikrocontroller und einem Bus, bei welchem der Mikroprozessor oder Mikrocontroller sowohl den Bus als auch den Motor steuert. Ein solcher Motor ist durch die Reduzierung der elektronischen Bauteile sehr billig, und weiterhin wird durch die Vermeidung weiterer elektronischer Bauteile eine kompakte Bauweise des Motors ermöglicht. Durch die Verlagerung der gesamten Steuerung des Motors in den Mikroprozessor oder Mikrocontroller sind Änderungen am Motor allein durch Änderungen der Software möglich.

Weitere Einzelheiten und vorteilhafte Weiterbildungen der Erfindung ergeben sich aus den im folgenden beschriebenen und in den Zeichnungen dargestellten, in keiner Weise als Einschränkung der Erfindung zu verstehenden Ausführungsbeispielen, sowie aus den übrigen Unteransprüchen. Es zeigt:

Fig. 1 ein Prinzipschaltbild einer erfindungsgemäßen Anordnung,

Fig. 2 eine beispielhafte Darstellung einer Ausführungsform der Erfindung,

Fig. 3 die Pinbelegung des μC COP 842 CJ,

Fig. 4 ein Schaltbild, welches die Bauteile für die A/D-Wandlung und die Verarbeitung des Hallsignals zeigt,

Fig. 5 ein Flußdiagramm der A/D-Wandlung,

Fig. 6 den Aufbau eines für die A/D-Wandlung verwendeten Zählers,

Fig. 7 ein Zeitdiagramm zum Ablauf der A/D-Wandlung,

Fig. 8 eine Kennlinie mit Hysteresis,

Fig. 9 ein Flußdiagramm für eine Hysteresefunktion,

Fig. 10 ein Flußdiagramm für eine Sensorabrisfunktion,

Fig. 11 eine Beispieldefinition für eine Kennlinie mit vier Definitionspunkten,

Fig. 12 eine Tabelle der Definitionspunkte der Kennlinie aus Fig. 11,

Fig. 13 ein Flußdiagramm für die Berechnung der Soll-drehzahl aus der Kennliniendefinition,

Fig. 14 eine Kennlinie mit einem für eine bestimmte Temperatur interpolierten Punkt,

Fig. 15 ein Flußdiagramm mit einer Variante der Berechnung der Solldrehzahl aus der Kennliniendefinition,

Fig. 16 eine Darstellung eines Hallsignals und zugeordneter Motorsignale,

Fig. 17 ein Flußdiagramm der Hall-Interruptroutine,

Fig. 18 ein Schaltbild mit für die Ansteuerung eines EEPROMS und den Zugang über einen Bus wichtigen Teilen,

Fig. 19 ein Schaltbild mit für die Regelung und den Antrieb des Elektromotors wichtigen Teilen,

Fig. 20 ein Flußdiagramm für den prinzipiellen Ablauf der Regelung der Drehzahl,

Fig. 21 ein Flußdiagramm der Berechnung des Stellwertes und des Vorzeichens für die Regelung aus Fig. 20,

Fig. 22A bis 22D eine Darstellung der Regelung für einen Motor mit der richtigen Drehzahl,

Fig. 23A bis 23E eine Darstellung der Regelung für einen Motor mit zu niedriger Drehzahl,

Fig. 24A bis 24E eine Darstellung der Regelung für einen Motor mit zu hoher Drehzahl,

Fig. 25 einen Gesamtablauf der Regelung,

Fig. 26 die Darstellung einer bevorzugten Ausführungsform eines Funktionsmanagers,

Fig. 27 ein in dem Funktionsmanager verwendetes Funktionsregister,

Fig. 28 eine für den Funktionsmanager modifizierte A/D-Wandlungsroutine und

Fig. 29 eine für den Funktionsmanager modifizierte Regelungsroutine.

Übersicht Kennlinienfunktion

Fig. 1 zeigt eine Prinzipdarstellung einer erfindungsgemäßen Anordnung. Ein Treiber 7 eines Motors 9 wird durch einen Drehzahlregler 6 gesteuert. Der Regler 6 erhält einen Istwert 8a der Drehzahl des Motors 9 von einem Tachogenerator 8, einen Sollwert 23a erhält er von einer Kennlinienfunktion 23. Die Kennlinienfunktion 23 berechnet, ausgehend von einer durch einen A/D-Wandler 10 digital gewandelten, ursprünglich analogen Größe 2, den Sollwert 23a für den Drehzahlregler 6 mit Hilfe von in einem Speicher 4 gespeicherten Stützwerten einer Kennlinie, wobei die nicht durch die Stützwerte vorgegebenen Werte durch Interpolation berechnet werden.

Übersicht Elektromotor

Fig. 2 zeigt eine Übersicht über ein bevorzugtes Ausführungsbeispiel eines erfindungsgemäßen Elektromotors. Dieser wird gesteuert von einem Mikrocontroller (μC) 11. Der Analog-Digital-Wandler (A/D-Wandler) 10 ist unter Verwendung eines im μC 11 befindlichen Komparators 20 aufgebaut und ermöglicht eine Digitalisierung der über einen NTC-Widerstand 18 detektierten Temperatur. Der Komparator 20 hat einen Minuseingang 21, der im folgenden auch als CP- oder CMPIN- bezeichnet wird, und er hat einen Pluseingang 22, der im folgenden auch als CP+ oder CMPIN+ bezeichnet wird. Diese Eingänge können durch

das Programm des μC 11 gesteuert werden, wie das nachfolgend beschrieben wird.

Ein konstanter Widerstand 16 ist in Reihe mit dem NTC-Widerstand 18 zwischen einer positiven Leitung 2 und Masse (GND) 100 angeschlossen. Ihr Verbindungspunkt 21a ist mit dem Minuseingang 21 verbunden.

Ebenso ist eine Konstantstromquelle 12 in Reihe mit einem Kondensator 14 zwischen der positiven Leitung 2 und Masse 100 angeschlossen, und ihr Verbindungspunkt 22a ist mit dem Pluseingang 22 des Komparators 20 verbunden.

Das Potential am Minuseingang 21 wird bestimmt durch die Temperatur am NTC-Widerstand 18, dessen Widerstandswert mit steigender Temperatur abnimmt, so daß dieses Potential mit steigender Temperatur sinkt.

Das Potential am Pluseingang 22 wird bestimmt durch die Spannung u_{C14} am Kondensator 14. Wenn der Pluseingang 22 programmgesteuert mit Masse 100 verbunden wird, wird $u_{C14} = 0$, und wenn anschließend der Pluseingang 22 auf einen hochohmigen Zustand ("Tristate") umgeschaltet wird, wird der Kondensator 14 über die Konstantstromquelle 12 mit einem Konstantstrom aufgeladen, so daß u_{C14} linear ansteigt.

Wenn das Potential am Punkt 22a das Potential am Punkt 21a erreicht hat, wird der Komparator 20 an seinem Ausgang 20a auf HIGH umgeschaltet. Die Zeit für die Ladung des Kondensators 14, ausgehend von $u_{C14} = 0$ V bis zum Umschalten des Ausgangs 20a auf HIGH ist also ein Maß für die Temperatur. Diese Zeit wird im μC 11 gemäß einer frei wählbaren Kennlinie umgesetzt in einen Sollwert für die Drehzahl des Motors 9.

Hierzu dient die Kennlinienfunktion 23. Diese bestimmt aus dem vom A/D-Wandler 10 digitalisierten Temperaturwert die Motor-Solldrehzahl 23a. Hierfür erhält sie über eine EEPROM-Funktion 24 Wertevorgaben aus einem nichtflüchtigen Speicher, hier einem EEPROM 26. Das EEPROM 26 kann über eine Kommunikationsfunktion 28 und ein Businterface 30 Werte für eine neue Kennlinie erhalten, um das Temperaturverhalten des Motors zu ändern.

Die Kennlinienfunktion 23 gibt die ermittelte Solldrehzahl 23a an den Drehzahlregler 6 weiter, welcher die Motorbestromung steuert. Dies kann z. B. über eine Steuerung eines Pulsweitenmodulation-Generators (PWM-Generators) 34 oder eine Blocksteuerung geschehen. Zu der Blocksteuerung wird beispielhaft auf DE 44 41 372.6 (intern: D183i) verwiesen.

Als einfaches Beispiel ist in Fig. 2 ein elektronisch kommutierter Motor 9 mit einer einzigen Phase 38 dargestellt. Die Bestromung dieser Phase 38 geschieht durch eine Transistorendstufe 36, hier in Form einer Vollbrücke 37. Ein Hallgenerator 40 liefert einer Antriebsfunktion 42 Information über die augenblickliche Stellung des Rotors 39. Die Antriebsfunktion 42 sorgt für die richtige Kommutierung des Motors 9 und den sicheren Betrieb, z. B. bei Überlastung des Motors 9.

Eine Strombegrenzung 44 vermindert die Bestromung der Endstufe 36, falls der Strom in der einzigen Phase 38 zu hoch wird, z. B. beim Start des Motors.

Am Ende der Beschreibung sind für die in den einzelnen Figuren verwendeten elektronischen Bauteile bevorzugte Werte angegeben. Der Leser wird hierauf verwiesen.

Fig. 3 zeigt die Pinbelegung des beim Ausführungsbeispiel verwendeten Mikrokontrollers (μC) 11 vom Typ COP 842 CJ der Firma National Semiconductors. Die Beschriftung innerhalb des μC 11 entspricht der Beschriftung des Herstellers, die äußere Beschriftung der jeweiligen Leitungen zeigt die hauptsächlich in der Anmeldung verwendeten Bezeichnungen. Zur Kennzeichnung der Lage ist links oben ein schwarzer Viertelkreis eingezeichnet, der sich in den fol-

genden Figuren wiederfindet.

Fig. 4 zeigt ein detailliertes Schaltbild des A/D-Wandlers 10 (Fig. 1 und 2) mit den Bauteilen für die A/D-Wandlung und die Verarbeitung des Hallsignals vom Hall-Sensor 40. Das Hallsignal liefert den Istwert der Motordrehzahl.

Ein Schwingquarz 97, der an die Anschlüsse CK0 und CK1 (vgl. Fig. 3) des μC 11 angeschlossen ist, gibt dessen Taktfrequenz vor, z. B. 10 MHz. Der Reset-Eingang Res (Fig. 3) ist über einen Kondensator 99 mit Masse 100 und über einen Widerstand 101 mit $+V_{cc}$ verbunden. Diese beiden Bauelemente erzeugen in der üblichen Weise beim Einschalten einen Power-Up-Reset.

Der Hallgenerator 40, z. B. vom Typ HW101A, ist zur Stromversorgung über einen Widerstand 106 mit $+V_{cc}$ und mit Masse 100 verbunden. Sein Ausgangssignal u_H wird den beiden Eingängen eines Komparators 108 (z. B. LM2901 D) zugeführt, dessen V_{cc} -Eingang ein Siebkondensator 110 zugeordnet ist. Der Ausgang des Komparators 108 ist über einen Rückführwiderstand 112 mit dem positiven Eingang des Komparators 108 und über einen sogenannten Pullup-Widerstand 114 mit $+V_{cc}$ verbunden. Weiterhin ist der Ausgang des Komparators 108 direkt mit dem Port Hall (Fig. 3) des Mikroprozessors 12 verbunden, so daß man an diesem ein vom Rotormagneten 39 gesteuertes Hallsignal erhält. Dieses Signal hat immer während einer Rotordrehung von 180° el. den Wert HALL = 0, und während der anschließenden Drehung von 180° el. den Wert HALL = 1. Seine Auswertung wird nachfolgend anhand von Fig. 17 erläutert. Jede Änderung von HALL = 1 zu HALL = 0, oder umgekehrt, bewirkt einen Interruptvorgang im μC 11.

Der NTC-Widerstand 18 des A/D-Wandlers 10 ist auf seiner einen Seite an Masse 100 angeschlossen. An seiner anderen Seite ist er mit dem Widerstand 16 verbunden, welcher auf seiner anderen Seite mit $+V_{cc}$ verbunden ist. Die Verbindung 21a zwischen dem NTC-Widerstand 18 und dem Widerstand 16 ist über einen Schutzwiderstand 89 und ein aus einem Kondensator 90 und einem Widerstand 91 bestehendes Siebglied an den Ausgang CP- (Fig. 3) des μC 11 angeschlossen. Der Kondensator 14 ist an seiner einen Seite mit Masse 100 verbunden, an seiner anderen Seite mit einem Widerstand 96, welcher seinerseits mit $+V_{cc}$ verbunden ist. Die Verbindung 22a zwischen dem Kondensator 14 und dem Widerstand 96 ist mit der Konstantstromquelle 12 und dem Eingang CP+ verbunden. Die Konstantstromquelle 12 weist einen npn-Transistor 95 (z. B. BC8568) auf, dessen Basisspannung durch Widerstände 92 und 93 festgelegt ist und dessen Strom auf der Emitterseite durch einen Widerstand 94 begrenzt wird.

A/D-Wandlung

Fig. 5 zeigt ein Flußdiagramm der A/D-Wandlung.

Im Schritt S100 der Fig. 5 wird ein Watchdog-Timer WDCNT 79 (Fig. 6) des μC 11 mit einem Hexadezimalwert 0xFF geladen (Hexadezimalwerte werden durch ein vorangestelltes 0x gekennzeichnet) und durch das Setzen des Bits WDREN auf 1 gestartet. Der Watchdog-Timer WDCNT 79 wird in einem Modus betrieben, in dem er in festen Zeitschritten seinen Wert dekrementiert und beim Erreichen von Null einen internen Reset im μC 11 auslöst. Um diesen Resetvorgang zu verhindern, muß der Watchdog-Timer WDCNT 79 von dem Programm periodisch nachgeladen werden. Dies bewirkt eine erhöhte Sicherheit, da der μC 11 z. B. bei einem Absturz des Programms nicht nachgeladen wird und deshalb automatisch einen Reset erhält und von neuem startet (Watchdog-Funktion).

In S102 werden die beiden Zählerregister CNT_LB 82 und CNT_HB 81 (Fig. 6) auf 0 gesetzt.

Die Eingänge CP- und CP+ des Komparators 20 (vgl. Fig. 2 und Fig. 3) werden in S104 konfiguriert. CP+ wird auf LOW (niedrig, Masse) gesetzt, so daß der Kondensator 20 entladen wird. CP- wird auf TRISTATE gesetzt, so daß eine von dem Widerstand 16 und NTC-Widerstand 18 bestimmte Spannung an CP- anliegt. Der Zustand TRISTATE eines Ports bedeutet, daß der Port sich weder wie +Vcc noch wie Masse 100 verhält, sondern isoliert ist.

Die mit dem Watchdog-Timer WDCNT 79 gebildete Hauptschleife beginnt in S106. Der Watchdog-Timer WDCNT 79 wird (wie in S100) mit dem Wert 0xFF geladen und durch WDREN := 1 gestartet. Nach dem Durchlaufen der im folgenden beschriebenen Schritte S108 bis S118 wird in S120 überprüft, ob der Watchdog-Timer WDCNT 79 bereits den Wert 0xFB erreicht hat, also viermal dekrementiert worden ist. Ist dies der Fall, so wird der aus den beiden Bytes CNT_LB 82 und CNT_HB 81 gebildete Zähler CNT 80, den man als (CNT_HB, CNT_LB) darstellen kann (vgl. Fig. 6), in S122 inkrementiert und zu S106 zurückgesprungen. War in S120 der Watchdog-Timer WDCNT 79 größer als 0xFB, so erfolgt ein Sprung zurück zu S108. Da der Watchdog-Timer 79 in 256 µs-Schritten dekrementiert, entsprechen diese vier Dekrement-Schritte bis 0xFB einer Zeit von 1024 µs. Der aus CNT_LB 82 und CNT_HB 81 gebildete Zähler CNT 80 hat also eine Auflösung von 1024 µs = 1,024 ms.

Fig. 6 zeigt hierzu schematisch den Aufbau des aus CNT_LB 82 und CNT_HB 81 gebildeten Zählers CNT 80. Die beiden 8-Bit-Zähler CNT_HB 81 und CNT_LB 82 werden zusammen als 16-Bit-Zähler CNT 80 verwendet. Hierbei ist CNT_LB 82 das niedrige Wort (low byte) und CNT_HB 81 das hohe Wort (high byte). Das niedrigste Bit jedes Bytes ist jeweils mit LSB bezeichnet, und das höchste Bit mit MSB. Jedes Byte hat acht Bits.

Der Watchdog-Timer WDCNT 79 erniedrigt sich z. B. alle 256 µs um 1. Durch die Frequenzteilung von z. B. 4 : 1 in S106 und S120, Fig. 5, wird der Zähler CNT 80 alle 1,024 ms inkrementiert. Dies entspricht einer Frequenz von ca. 1000 Hz. Findet bei dem Inkrementieren des CNT_LB 82 ein Überlauf statt, d. h. das CNT_LB 82 hat den Wert 0xFF und wird inkrementiert, so erhält das CNT_LB 82 den Wert 0x00, und ein Überlaufbit (carry bit) wird gesetzt. Nach dem Inkrement des CNT_LB 82 wird zum CNT_HB 81 eine Null unter Berücksichtigung des Überlaufbits addiert, d. h. bei jedem 256. Inkrement des CNT_LB 82 wird das CNT_HB 81 inkrementiert, und das CNT_HB 81 hat so eine Auflösung von ca. 256 ms und kann deshalb einen maximalen Zeitwert von ca. 65,5 s darstellen. Der Zähler CNT 80, der aus den beiden 8-Bit-Zählern CNT_HB 81 und CNT_LB 80 zusammengesetzt ist, wirkt so als ein 16-Bit-Zähler, der, jeweils nach 4 Dekrement-Schritten des Watchdog-Timers WDCNT 79, um 1 inkrementiert wird.

Obwohl CNT_LB 82 und CNT_HB 81 zusammen als ein Zähler CNT 80 wirken, haben sie mehrere unterschiedliche Funktionen:

- Das CNT_LB 82 dient bei der A/D-Wandlung als Zähler für die Zeit, die benötigt wird, um den Kondensator 14 über den Transistor 95 so weit aufzuladen, daß die Spannung an CP+ 22 so hoch ist wie die durch den NTC-Widerstand 18 temperaturabhängige Spannung an CP- 21, d. h. sein Zählerstand ist ein Maß für die Temperatur des NTC-Widerstands 18.
- Das CNT_HB 81 dient als Zähler für die Zeit zwischen aufeinanderfolgenden A/D-Wandlungen. Diese Zeit kann z. B. auf eine Sekunde eingestellt werden.
- Das niederwertigste Bit (LSB) 83 des CNT_HB 81 dient zusätzlich als Anzeige eines Überlaufs des

CNT_LB 81 während der A/D-Wandlung.

Im folgenden werden die in dem unteren Teil der Watchdog-Timer-Schleife (Fig. 5) ablaufenden Schritte S108 bis S120 beschrieben.

In S108 wird überprüft, ob das CNT_HB den Wert 0x00 hat. Falls dies der Fall ist, wird in S110 der vorige NTC-Wert NTC_VAL in NTC_OLD gesichert, das CNT_HB wird auf 0xFC gesetzt, CP+ 22 wird auf TRISTATE gesetzt, so daß der Kondensator 14 über den mittels der Widerstände 92, 93 und 94 als Konstantstromquelle geschalteten Transistor 95 (Fig. 4) geladen wird und so die Spannung an CP+ 22 linear ansteigen läßt, und der Komparator 20 wird durch CMPEN := 1 gestartet. Das Register CNT_LB hat nach dem Umspringen von CNT_HB auf 0x00 ebenfalls den Wert 0x00, so daß es als Zähler für die Zeit wirkt, bis die an CP+ 22 liegende Spannung so groß wie die an CP- 21 liegende Spannung ist.

Das Setzen des Registers CNT_HB auf 0xFC gibt die Zeit zwischen den einzelnen A/D-Wandlungen vor. Da das Register CNT_HB eine Auflösung von ca. 256 ms hat, wird jeweils nach vier Inkrementen (0xFC auf 0xFD auf 0xFE auf 0xFF auf 0x00), also nach ungefähr einer Sekunde, eine neue A/D-Wandlung vorgenommen. Anstatt 0xFC kann ein anderer Wert gewählt werden, er muß aber größer als 0x00 sein, um nicht sofort eine neue A/D-Wandlung zu starten, und sein niederwertigste Bit 83 (Fig. 6) muß 0 sein, um es als Anzeige eines Überlaufs des CNT_LB 82 während der A/D-Wandlung nutzen zu können.

Bis die durch den von der Konstantstromquelle 12 (Fig. 4) geladenen Kondensator 14 gegebene Spannung an CP+ 22 die durch den NTC-Widerstand 18 temperaturabhängige Spannung an CP- 21 erreicht hat, wird in Fig. 5 der untere Teil über S108 und S112 durchlaufen, ohne weitere Schritte auszuführen. Erreicht die Spannung an CP+ 22 die Spannung an CP- 21, so schaltet der Komparator sein Komparator-Lesebit (CMPReadBit) auf HIGH, und von dem Vergleich in S112 wird zu S114 gesprungen.

In S114 wird anhand des niederwertigsten Bits 83 (LSB) von CNT_HB 81 überprüft, ob während der A/D-Wandlung ein Überlauf des CNT_LB 82 stattgefunden hat. Der Wertebereich 0x00 bis 0xFF des CNT_LB 82 soll möglichst vollständig für die D/A-Wandlung ausgenutzt werden. Deshalb kann es innerhalb der Fertigungstoleranzen der Komparatorbeschaltung vorkommen, daß für Werte im oberen Bereich, also für niedrige Temperaturen, noch ein Überlauf stattfindet. Dieser kann deshalb erkannt werden, weil das niederwertigste Bit 83 beim Setzen des CNT_HB 81 in S110 den Wert 0 erhalten hat. Hat ein Überlauf des CNT_LB stattgefunden, so hat das niederwertigste Bit 83 des CNT_HB 81 den Wert 1. Ist dies der Fall, so erhält CNT_LB in S116 den maximalen Wert 0xFF zugewiesen, ansonsten erfolgt von S114 direkt ein Sprung zu S118.

In S118 wird der Wert von CNT_LB invertiert und im Register NTC_VAL abgelegt. Das Invertieren macht aus CNT_LB den Wert (255-CNT_LB), so daß jetzt ein kleiner NTC-Wert NTC_VAL einer niedrigen Temperatur und ein großer NTC-Wert NTC_VAL einer hohen Temperatur entspricht. Der Komparator wird durch CMPEN := 0 gestoppt und CP+ 22 wird auf LOW gesetzt, damit der Kondensator 14 vor der nächsten A/D-Wandlung entladen wird. Letztere findet statt, wenn das CNT_HB den Wert Null erreicht.

Fig. 7 zeigt ein Zeitdiagramm der A/D-Wandlung. CNT_HB 81 dient als Zähler für die Zeit zwischen den einzelnen A/D-Wandlungen. Es hat am Anfang den Wert 0xFF, und CP+ ist auf LOW, der Kondensator 14 ist also entladen.

Zum Zeitpunkt 84 bzw. 84' springt CNT_HB 81 auf 0x00 um. Damit startet die A/D-Wandlung. CNT_HB 81 wird auf

0xFC gesetzt, um den Zeitraum bis zur nächsten A/D-Wandlung festzulegen, und CP+ wird auf TRISTATE gesetzt, um ein Aufladen des Kondensators 14 zu ermöglichen.

Zum Zeitpunkt 85 bzw. 85' erreicht die Spannung u_{C14} des Kondensators 14 das Potential 86 am Knotenpunkt 21a, und die Variable CMPReadBit ändert ihren Zustand von 0 auf 1. Jetzt wird geprüft, ob ein Überlauf von CNT_LB 82 stattgefunden hat, der Komparator 20 (Fig. 2) wird ausgeschaltet, und CP+ wird auf LOW geschaltet, um den Kondensator 14 wieder zu entladen. Die nächste A/D-Wandlung geschieht beim Inkrement des CNT_HB von 0xFF auf 0x00, also zum Zeitpunkt 84'.

Hysteres- und Sensorabrißfunktion

Der eigentlichen Berechnung der Solldrehzahl durch die Kennlinienfunktion sind eine Hysteres- und eine Sensorabrißfunktion vorgeschaltet.

Fig. 8 zeigt eine Kennlinie 180 (durchgezogene dünne Kurve) mit Hysteres 182 (gestrichelte dünne Kurve), in die ein Beispielverlauf 184 von Temperatur und Solldrehzahl eingetragen ist (dicke durchgezogene Kurve).

Die Solldrehzahl n_s verläuft bei zunehmender Temperatur T von dem Punkt 185 entlang der durchgezogenen Kennlinie 180 bis zum Punkt 186.

Bei abnehmender Temperatur bleibt die Solldrehzahl so lange konstant, bis die gestrichelte Hysteresekurve bei Punkt 187 erreicht wird. Daraufhin springt die Solldrehzahl auf die für diese Temperatur durch die durchgezogene Kennlinie vorgegebene Solldrehzahl bei Punkt 188.

Bei ansteigender Temperatur verläuft die Solldrehzahl entlang der Kennlinie 180 vom Punkt 188 zum Punkt 189, an dem der Beispielverlauf 184 endet. Die Hysteres wirkt so einem Schwingen der Drehzahl entgegen, weil bei abnehmender Temperatur die Drehzahl erst dann reduziert wird, wenn die Temperatur um einen bestimmten Mindestwert gefallen ist.

Fig. 9 zeigt das Hysteres-Flußdiagramm. Die Hysteres wird dadurch erreicht, daß der vom A/D-Wandler digitalisierte Temperaturwert NTC_VAL mit dem vorherigen und zuvor abgespeicherten Wert NTC_OLD verglichen wird. Ist NTC_VAL größer als NTC_OLD, so wird er für die Berechnung der Solldrehzahl n_s verwendet (S210), weil die Temperatur zugenommen hat. Ist die Antwort "Nein", so wird in Schritt S212 NTC_VAL von NTC_OLD subtrahiert. Ist die Differenz größer als ein Hysteresewert HYST, so wird die Hystereseroutine verlassen und mit dem Wert NTC_VAL weitergearbeitet (S212). Ansonsten wird dem Wert NTC_VAL der alte Wert NTC_OLD zugeordnet (S214). Der Hysteresewert HYST kann bei der Konfiguration des Motors in den RAM-Bereich des $\mu C 11$ geladen werden und ist somit ebenfalls frei programmierbar. Der Motor erhält durch die Hysteres-Funktion einen ruhigen Drehzahlverlauf.

Fig. 10 zeigt ein Flußdiagramm einer Sensorabrißfunktion. Die Sensorabrißfunktion ist eine Sicherheitsfunktion, die bei einer Beschädigung des NTC-Widerstands 18 (Fig. 4) bzw. von dessen Anschlüssen, einem sogenannten Sensorabriß, eine sogenannte Sensorabriß-Drehzahl als Solldrehzahl vorgibt. Ist z. B. der NTC-Widerstand defekt, und hat er den Widerstandswert Unendlich, so erhält man einen sehr kleinen NTC-Wert NTC_VAL. Daher wird ein Sensorabriß-Temperaturwert T_{SA} definiert, und bei allen NTC-Werten NTC_VAL, die kleiner als T_{SA} sind, wird davon ausgegangen, daß ein Sensorabriß vorliegt.

Der aus der zuvor beschriebenen A/D-Wandlung erhaltene NTC-Wert NTC_VAL wird in S200 mit dem Sensorabriß-Temperaturwert T_{SA} , der ebenfalls 1 Byte groß ist

(z. B. $T_{SA} := 0 \times 38$), verglichen. Ist NTC_VAL größer, so wird die Berechnung der Solldrehzahl n_s wie in den folgenden Figuren beschrieben fortgesetzt (S202). Ist NTC_VAL jedoch kleiner, liegt also ein Sensorabriß vor, so wird die komplette Solldrehzahlberechnung übersprungen, und für die Solldrehzahl n_s wird eine Sensorabriß-Drehzahl n_{SA} verwendet (S204), gewöhnlich die maximale Drehzahl des Motors.

Die Werte T_{SA} und n_{SA} werden bei der Konfiguration des Motors in den RAM-Bereich des $\mu C 11$ geladen und sind somit frei programmierbar.

Kennlinienfunktion

Der durch die A/D-Wandlung ermittelte NTC-Wert NTC_VAL wird nun in eine Solldrehzahl umgerechnet.

Fig. 11 zeigt beispielhaft eine durch vier Kennliniendefinitionspunkte (ausgefüllte Punkte mit den Zahlen 1 bis 4) definierte Kennlinie $n = f(T)$. Z. B. entspricht eine Temperatur von 0°C (Punkt 1) einer Drehzahl von 2000 U/min. ebenso eine Temperatur von 30°C (Punkt 2). Eine Temperatur von 60°C (Punkt 3) entspricht 4000 U/min. ebenso eine Temperatur von 100°C (Punkt 4). Zwischen den Temperatur-Drehzahl-Punkten 1, 2, 3 und 4 wird bei diesem Ausführungsbeispiel linear interpoliert. Die Berechnung der benötigten Zwischenpunkte geschieht durch den $\mu C 11$.

Die Definition einer Kennlinie durch wenige Kennliniendefinitionspunkte spart viel Speicher und erlaubt eine einfache Änderung der Kennlinie durch Speicherung von Punkten mit neuen Werten. Falls eine Sensorabrißfunktion verwendet wird, wird der Definitionspunkt mit der niedrigsten Temperatur an der auf die Sensorabrißtemperatur T_{SA} (vgl. S200 in Fig. 10) folgenden Temperatur $T_{SA} + 1$ gewählt. Falls keine Sensorabrißfunktion verwendet wird, wird der Definitionspunkt mit der niedrigsten Temperatur als kleinste meßbare Temperatur (Kennlinientemperaturwert 0x00, entspricht z. B. einer Temperatur von -62°C) gewählt.

Der Definitionspunkt mit der höchsten Temperatur kann als die höchstmögliche Temperatur gewählt werden (0xFF bei einem Byte Speicherplatz). Eine Alternative besteht darin, für alle Temperaturwerte, die größer als der letzte Definitionspunkt sind, die Drehzahl des letzten Definitionspunkts zu wählen. Dann kann der letzte Definitionspunkt auch einen niedrigeren Temperaturwert als 0xFF haben.

Fig. 12 zeigt die für jeden Kennliniendefinitionspunkt P der Kennlinie aus Fig. 11 die gespeicherten Temperatur- und Drehzahlwerte. Die Werte sind jeweils in physikalischen und programminternen Größen angegeben. Jeder Temperaturwert T belegt ein Byte Speicher, der zugehörige Drehzahlwert n zwei Byte. Evtl. wird noch eine zugehörige Steigung $S = n/T$ mit zwei Byte gespeichert, um die Interpolation zu beschleunigen. Das höchstwertigste Bit der zwei Bytes der Steigung dient dabei als Vorzeichen-Bit, so daß auch negative Steigungen möglich sind. Die Zuordnung der programminternen Temperaturspeicherwerte (0x00 bis 0xFF) zu der physikalischen Außentemperatur hängt von der Beschaltung des A/D-Wandlers ab. So kann z. B. ein Temperaturspeicherwert von 0x10 auch einer negativen physikalischen Temperatur von -10°C entsprechen.

Fig. 13 zeigt ein Flußdiagramm für die Solldrehzahl-Berechnungsroutine. Die Kennliniendefinitionspunkte besitzen in diesem Beispiel jeweils einen Temperaturwert, einen Drehzahlwert und eine Steigung, die für den Bereich von dem Kennliniendefinitionspunkt bis zu dem folgenden Kennliniendefinitionspunkt gilt. Beim Starten des $\mu C 11$ werden die Temperaturwerte T (Fig. 12) aus dem EEPROM 26 ausgelesen und in den RAM-Bereich des $\mu C 11$ eingele-

sen, um einen schnellen Zugriff zu ermöglichen. Die Drehzahlwerte und die Steigungswerte verbleiben (aus Gründen des begrenzten RAM-Speichers) im EEPROM 26 und werden bei Bedarf von dort geladen. Im EEPROM 26 sind alle Temperaturwerte hintereinander in einem Block gespeichert, die Drehzahl- und gegebenenfalls Steigungswerte hintereinander in einem anderen Block. Falls genügend RAM-Speicher vorhanden ist, so können auch die Drehzahl und die Steigung in den RAM-Speicher des μC 11 geladen werden.

Beim Sprung in die Solldrehzahl-Berechnungsroutine wird in S300 der Zähler N auf 1 gesetzt. Die Temperaturwerte sind als Tabelle T(N) abgelegt, und in S302 wird verglichen, ob der N-te Temperaturwert T(N) kleiner als NTC_VAL ist. Ist dies der Fall, so wird nach S301 gesprungen, und N wird um 1 erhöht. Daraufhin wird wieder nach S302 gesprungen. Ist NTC_VAL nicht mehr größer als T(N), so weiß man, daß entweder $\text{NTC_VAL} = T(N)$ ist, oder zwischen T(N) und T(N-1) liegt. In S304 wird auf Gleichheit geprüft. Besteht Gleichheit, so wird der Solldrehzahl n_s der Solldrehzahlwert n(N) des Kennliniendefinitionspunktes N zugeordnet und an das Ende gesprungen.

Ist in S304 nicht $T(N) = \text{NTC_VAL}$, so wird in S306 der Variablen A der Wert $N - 1$ zugeordnet, A entspricht also dem dem Kennliniendefinitionspunkt N vorhergehenden Kennliniendefinitionspunkt.

In S307 wird dann schließlich die Solldrehzahl n_s für NTC_VAL berechnet.

Fig. 14 zeigt hierzu ein Beispiel für einen Wert NTC_VAL 159, der zwischen dem zweiten und dem dritten Definitionspunkt einer Kennlinie liegt. A hat also den Wert 2.

Der Abstand X zwischen dem Temperaturwert T(A) und NTC_VAL wird durch Subtraktion ermittelt (S307). Die zu NTC_VAL zugehörige Solldrehzahl n_s wird durch Addition des Produkts aus Steigung S(N) und X zu der Drehzahl n(A) von Punkt A ermittelt, also $n_s := n(A) + X \cdot S(N)$.

Eine mögliche Variante zu dieser Art der Interpolation besteht darin, die Steigung nach links, also von dem jeweiligen Kennliniendefinitionspunkt N in Richtung des Kennliniendefinitionspunktes N-1, zu definieren. In diesem Fall geschieht die Berechnung von n_s (S306, S307) nicht von Punkt N-1 aus, sondern von Punkt N.

Fig. 15 zeigt die Berechnung der Solldrehzahl für den Fall, daß bei den Kennliniendefinitionspunkten keine Steigung mit abgespeichert ist. Die Schritte S300 bis S305 aus Fig. 13 werden identisch ausgeführt. Falls in S304 jedoch nicht $T(N) = \text{NTC_VAL}$ ist, so werden in S306' beide den NTC_VAL 159 (Fig. 14) umgebenden Kennliniendefinitionspunkte A := N - 1 und B := N definiert, siehe wieder Fig. 14. In S307 wird nun die Steigung S zwischen Punkt A und B von Punkt A aus berechnet. Die Temperaturdifferenz D_T (T) wird durch Subtraktion von T(A) von T(B) berechnet, die Drehzahldifferenz D_n (n) durch Subtraktion von n(A) von n(B) und die Steigung S durch Division von D_n durch D_T. Der Abstand X zwischen NTC_VAL und dem Temperaturwert T(A) und die Solldrehzahl n_s werden wie in Fig. 13 berechnet.

Aus der Beschreibung von Fig. 15 ist ersichtlich, daß das Abspeichern einer Steigung zu jedem Kennliniendefinitionspunkt einen Teil der Berechnungen in S307 (Fig. 13) erspart und damit auch Programm-Code und Programm-Abarbeitungszeit.

Umrechnung der Solldrehzahl in eine "Hall-Länge" HL_s

Fig. 16 zeigt ein Diagramm mit dem Hallsignal HALL (Fig. 4), das am Port INT des μC 11 (Fig. 4) anliegt, den zu-

gehörigen Werten für OUT1, OUT2 (Fig. 19) und den Zeitpunkten für das Auftreten der Hall-Interrupts. Anhand des Signals HALL wird die Kommutierung der Signale OUT1 und OUT2 (Fig. 19) für die Steuerung des Motors 9 und die Berechnung der Ist-Hall-Länge HL_i vorgenommen, wie es hier in einer beispielhaften Ausführung gezeigt wird. Jeder Wechsel von HALL löst in dem μC 11 einen Hall-Interrupt aus, welcher in der Fig. 16 durch ein Y gekennzeichnet ist. Da die Kommutierung sehr präzise sein muß, damit der Motor ruhig läuft, hat dieser Interrupt Vorrang vor allen anderen Vorgängen im Motor, d. h. er unterbricht alle anderen, gerade laufenden Prozesse, wobei aber der gerade in Arbeit befindliche Befehl noch ausgeführt wird, anschließend die Befehlsfolge dieses Interrupts abläuft, und dann der zuvor unterbrochene Prozess wieder fortgesetzt wird.

Fig. 17 zeigt eine beispielhafte Hall-Interruptroutine, die bei jedem Hall-Interrupt ausgeführt wird.

In S320 wird die Hall-Länge HL_i bestimmt. Ein aktueller Timerwert t_A wird aus einem Timer ausgelesen, und durch die Subtraktion des gespeicherten Timerwerts t_O vom Zeitpunkt des vorhergehenden Timer-Interrupts wird die Hall-Länge $\text{HL}_i := t_A - t_O$ berechnet. Daraufhin wird der aktuelle Timerwert t_A in t_O gespeichert (S320). Die Auflösung des in diesem Ausführungsbeispiels verwendeten Timers ist 1 μs , die Hall-Länge HL_i liegt also in μs vor. Sie ist ein Maß für die Drehzahl des Motors 9.

In den folgenden Schritten wird die Kommutierung ausgeführt. In S322 wird geprüft, ob HALL = 1 (HIGH) ist. Ist HALL = 1, so wird in S324 OUT2 auf LOW gesetzt. Jetzt sind OUT1 und OUT2 auf LOW, und in S326 wird eine zeitliche Kommutierungslücke eingefügt, um bei der Kommutierung einen Kurzschluß in der Brückenschaltung 37 zu verhindern. Die Kommutierungslücke hat z. B. eine Dauer von 50 μs . In S328 wird OUT1 auf HIGH gesetzt. In S329 wird schließlich der Port HALL konfiguriert, bei welcher Flanke er einen Hall-Interrupt HALL_INT auslösen soll. Die Flanke kann entweder so eingestellt werden, daß bei dem Übergang HIGH nach LOW (fallende Flanke) ein Interrupt ausgelöst wird, oder aber bei dem Übergang von LOW nach HIGH (steigende Flanke). Da in dem Zweig S324 bis S329 das Hall-Signal auf HIGH ist, muß der Port HALL auf einen Interrupt bei fallender Flanke, also HIGH nach LOW eingestellt werden, damit beim nächsten Hall-Wechsel wieder ein Hall-Interrupt ausgelöst wird.

Ist in S322 HALL = 0 (LOW), so geschieht in S330-S335 analog die umgekehrte Kommutierung und das umgekehrte Setzen von HALL_INT. Für die Realisierung einer zeitlichen Verschiebung der Kommutierungszeitpunkte wird auf DE 197 00 479.2 (intern: D201i) verwiesen.

Die Umrechnung der Solldrehzahl n_s mit der Einheit U/min in eine "Hall-Länge" HL_s wird im folgenden aufgezeigt. Für die Hall-Länge HL' in Sekunden gilt

$$\text{HL}' = T/P$$

mit der Periodendauer T für eine Rotordrehung in s und der Polzahl P des Rotors. Mit

$$T = 1/f$$

$$f = n/60$$

wobei f die Frequenz in Hz und n die Drehzahl in Umdrehungen pro Minute ist, ergibt sich

$$\text{HL}' = 60[s]/(n[\text{min}^{-1}]P)$$

Da die über den Hall-Sensor gemessene Hall-Länge in μs vorliegt, wird HL' zu HL_s renormiert

HL_s = 1 000 000 HL'

Für P = 4, also einen vierpoligen Rotor, ergibt sich

HL_s = 15 000 000 [µs]/(n[min^{-1}]).

Der gewünschten Drehzahl n_s = 2870 min^{-1} entspricht z. B. eine "Hall-Länge" HL_s von

HL_s = 15 000 000 µs/2870 = 5226 µs.

Die maschineninterne Hexadezimaldarstellung hierfür ist 0x146A. Zur Auswertung von HL_i und HL_s vergleiche Fig. 20.

EEPROM-Funktion

Fig. 18 zeigt den Ausschnitt der Schaltung, der das EEPROM 26 und das Businterface 30 betrifft. Die Pinbelegung des µC 11 ist wieder Fig. 3 zu entnehmen. Gleiche oder gleich wirkende Teile wie in den vorhergehenden Figuren sind mit denselben Bezugszeichen bezeichnet wie dort. Das EEPROM 26 ist z. B. vom Typ "2-Wire Serial CMOS EEPROM AT24C01A" (ATMEL).

Das EEPROM 26 erhält an seinem Dateneingang SDA das Signal ESDA (Fig. 3) des µC 11 und an seinem Eingang SCL das Signal ESCL. Beide Leitungen sind über Widerstände 172, 173 mit +Vcc verbunden.

Der Schreibschutzzeigang WP des EEPROM 26 ist mit dem Pin CS (Chip Select) des µC 11 verbunden. Ist CS auf HIGH, so ist das EEPROM 26 schreibgeschützt, ist CS auf LOW, so können Daten in das EEPROM 26 geschrieben werden. Die Anschlüsse VSS, AD, A1 und A2 des EEPROM 26 sind mit Masse 100 verbunden, und der Anschluß VCC des EEPROM 26 mit +Vcc.

Die Leitungen ESDA und ESCL stellen also den seriellen Bus zwischen dem µC 11 und dem EEPROM 26 dar, der als IIC-Bus betrieben werden kann.

Normalerweise wird das EEPROM 26 in der Fabrik einmal über das Businterface 30 programmiert, eine Neuprogrammierung ist jedoch jederzeit möglich. Auch kann das EEPROM 26 z. B. als Betriebsdatenspeicher dienen, z. B. für Einschaltzyklen, maximal aufgetretene Temperatur, Betriebsstunden und Fertigungsdaten.

Das Businterface 30 arbeitet mit einem IIC-Bus. Es verfügt über eine Datenleitung DATA mit einem Anschluß 160, die über einen Widerstand 162 an den Anschluß SDA des µC 11 angeschlossen ist. Vom Anschluß SDA führt ein Widerstand 165 zu +Vcc und ein Kondensator 167 zu Masse 100. Außerdem ist der Anschluß SDA mit dem Emitter eines npn-Transistors 168 verbunden, dessen Kollektor mit Masse 100 und dessen Basis über einen Widerstand 169 mit dem Anschluß N16 des µC 11 verbunden sind.

Weiterhin hat das Businterface 30 eine Taktleitung CLOCK mit einem Anschluß 161, die über einen Widerstand 163 an den Anschluß SCL des µC 11 angeschlossen ist. Vom Anschluß SCL des µC 11 führt ein Widerstand 164 zu +Vcc und ein Kondensator 166 zu Masse 100.

Die Schaltung mit dem npn-Transistor 168 dient dazu, sowohl den Ausgang N16 als auch den Eingang SCL des µC 11 mit der bidirektionalen Leitung DATA des IIC-Buses zu verbinden.

Für eine weitergehende Beschreibung des EEPROM 26, des Businterfaces 30 und deren Programmierung wird auf DE 198 26 458.5 (intern: D215) verwiesen.

Ein Paßwortschutz kann dadurch verwirklicht werden, daß bei der Übertragung von zwei besonderen aufeinander-

folgenden Bytes von dem externen IIC-Bus 30 zu dem µC 11, z. B. den Bytes 0xFA und 0x4A, in dem µC 11 ein Bit B ACCESS in einer Speicherstelle ACCESS gesetzt wird, und bei gesetztem B_ACCESS zusätzliche Funktionen zum Verändern von Parametern, zum Auslesen des EEPROM 26 und/oder zum Beschreiben des EEPROM 26 zur Verfügung stehen. So ist z. B. ein Schutz vor einem unerlaubten Ändern der Kennlinie durch den Kunden möglich.

Weiterhin kann über den Bus 30 ein Betriebsstundenzähler abgefragt werden. Dieser wird dadurch verwirklicht, daß beim Start des Lüfters ein aus drei Byte bestehender 24-Bit-Wert WRK_TIME aus dem EEPROM 26 in den µC 11 geladen wird, und bei jedem 10. Beginn einer A/D-Wandlung, also in dem vorherigen Beispiel jede 10. Minute, der 24-Bit-Zähler WRK_TIME um 1 inkrementiert und in das EEPROM 26 zurückgeschrieben wird. Der Wert WRK_TIME kann dann z. B. über den IIC-Bus 30 abgefragt werden. Das Abzählen von je 10 A/D-Wandlungen geschieht über einen weiteren Zähler WRK_10.

Regelung des Motors

Fig. 19 zeigt den für Regelung und Antrieb des Motors wichtigen Teil der Schaltung. Die Belegung der Anschlüsse des µC 11 ist wieder Fig. 3 zu entnehmen. Die Ausgänge OUT1 und OUT2 des µC 11 steuern die als H-Brücke 37 geschalteten npn-Transistoren 141, 142, 143 und 144. Je nachdem, ob OUT1 auf HIGH und OUT2 auf LOW oder umgekehrt gesetzt sind, verläuft der Strom durch die Statorwicklung 38 in die eine oder die andere Richtung. Zwischen dem Umschalten sind OUT1 und OUT2 kurzzeitig beide auf LOW, um einen Kurzschluß in der Brücke 37 verhindern. Die Kommutierung erfolgt elektronisch, und die Lage des Rotors 39 wird über den Hallsensor 40 erfaßt, der bei Fig. 4 näher beschrieben ist.

Ein Ausgang RGL des µC 11 ist über einen Widerstand 123 an einen Kondensator 124 angeschlossen. Wird RGL auf HIGH gesetzt, so wird der Kondensator 124 aufgeladen, ist RGL auf LOW, so wird der Kondensator entladen, und ist RGL auf TRISTATE, so ist der Kondensator 124 von RGL abgekoppelt und hält seine Spannung. Ohne die Strombegrenzung 44, welche weiter unten beschrieben wird, könnte der Punkt 125 direkt mit dem Pluseingang des Komparators 120 verbunden werden.

Ist der npn-Transistor 150 nicht leitend, die Strombegrenzung 44 also inaktiv, so stellt sich über den Widerstand 126 an einem kleineren Kondensator 127 die gleiche Spannung wie die des Kondensators 124 ein. Über den Ausgang RGL des µC 11 kann also die Spannung an dem Pluseingang des Komparators 120 beeinflusst werden.

An dem Minuseingang des Komparators 120 liegt ein durch einen Dreiecksoszillator 35 erzeugtes Dreieckssignal an. Der Dreiecksoszillator 35 weist einen Komparator 130 auf. Vom Ausgang P3 des Komparators 130 führt ein Mitkopplungswiderstand 132 zu dessen Pluseingang, und ebenso führt ein Gegenkopplungswiderstand 131 vom Ausgang P3 des Komparators 130 zum Minuseingang des Komparators 130. Ein Kondensator 135 liegt zwischen dem Minuseingang des Komparators 130 und Masse 100. Der Ausgang des Komparators 130 ist ferner über einen Widerstand 133 mit +Vcc verbunden. Der Pluseingang des Komparators 130 ist über zwei Widerstände 134 und 136 mit +Vcc bzw. Masse 100 verbunden.

Zur Erläuterung der Wirkungsweise des Dreiecksgenerators 35 sind in Fig. 19 drei Potentialpunkte P1, P2 und P3 angegeben. Beim Einschalten der Anordnung liegt P1 durch den entladenen Kondensator 135 auf Masse 100, P2 liegt über 134 an +Vcc und ist damit größer als P1. Der Kompa-

ratorausgang und damit P3 liegen deshalb auf HIGH. Dadurch wird der Kondensator 135 über die Widerstände 133 und 131 geladen, das Potential an P1 und damit das Dreieckssignal steigen an. Der Wert P2 ergibt sich aus

- a) der Parallelschaltung der Widerstände 134, 133 und 132 und
- b) dem unteren Spannungsteilerwiderstand 136.

Durch das Aufladen des Kondensators 135 wird schließlich P1 höher als P2, und dadurch schaltet der Ausgang P3 des Komparators 130 auf LOW, also Masse um. P3 geht also auf Null. Deshalb beginnt nun der Kondensator 135, sich über den Widerstand 131 und den Komparator 130 zu entladen, und dies ergibt den abfallenden Teil des Dreieckssignales. Der Wert P2 ergibt sich nun aus

- a) der Parallelschaltung der Widerstände 132 und 136 und
- b) dem Spannungsteilerwiderstand 134.

Fällt durch die Entladung des Kondensators 130 P1 unter P2, so schaltet der Komparator 130 wieder auf HIGH, an P3 liegt also wieder +Vcc. So entsteht ein Dreieckssignal mit z. B. 25 kHz.

Liegt die Spannung des Dreieckssignals am Minuseingang des Komparators 120 unter der des Referenzsignals am Pluseingang des Komparators 120, so ist der Ausgang OFF des Komparators 120 auf HIGH, und die unteren Transistoren 141 bzw. 143 können über die logischen UND-Glieder 147 bzw. 148 durch OUT1 bzw. OUT2 geschaltet werden. Liegt die Spannung des Dreieckssignals über der des Referenzsignals, so ist der Ausgang OFF des Komparators 120 auf LOW und damit kann die Statorwicklung 38 nicht bestromt werden.

Über die Spannung am Kondensator 124 und damit auch am Kondensator 127 wird also das sogenannte Tastverhältnis eingestellt, das Verhältnis der Dauer, die der Ausgang des Komparators 120 während einer Periode des Dreieckssignals auf HIGH ist zu einer ganzen Periode. Das Tastverhältnis kann zwischen 0% und 100% liegen. Ist die Motordrehzahl z. B. zu hoch, so wird der Kondensator 124 über RGL entladen und damit das Tastverhältnis verkleinert. Das Ganze wird als Pulsweitenmodulation (PWM) bezeichnet. Der Pull-Up-Widerstand 128 dient dazu, den Open-Collector-Ausgang OFF des Komparators 120 bei HIGH auf +Vcc zu ziehen.

Um beim Einschalten des Motors den Motor starten zu können, wird der Kondensator 124 bei der Initialisierung eine vorgegebene Zeitdauer über RGL aufgeladen, damit die Spannung am Kondensator 127 den erforderlichen Mindestwert für das Einschalten der Brücke 37 erreicht.

Die Strombegrenzung 44 wird dadurch verwirklicht, daß der Strom in der Statorwicklung 38 über einen Meßwiderstand 140 zu Masse 100 fließt. Je höher der Strom durch den Widerstand 140 ist, desto höher ist die Spannung an ihm und damit auch das Potential am Punkt 149.

Erreicht das Potential an 149 einen bestimmten Wert, so wird der Transistor 150 leitend und reduziert die Spannung am Kondensator 127, und das Tastverhältnis am Ausgang des Komparators 120 wird dadurch kleiner. Der Widerstand 126 verhindert, daß der große Kondensator 124 bei einer Strombegrenzung mit entladen und beschleunigt die Strombegrenzung, da der kleine Kondensator 127 schnell entladen werden kann. Nach dem Ende der aktiven Strombegrenzung wird der kleinere Kondensator 127 durch den Kondensator 124 wieder aufgeladen und so auf dessen Spannung gesetzt. Der Widerstand 126 und der Kondensator

127 bewirken also eine Prioritierung der Strombegrenzung 44.

Die Strombegrenzung 44 weist ein Filterglied aus einem Widerstand 151 und einem Kondensator 152 gegen Masse, gefolgt von dem npn-Transistor 150, der bei einer genügend hohen Spannung an seiner Basis den Pluseingang des Komparators 120 auf Masse 100 zieht, auf. Dahinter folgt ein weiteres aus den Widerständen 153 und 155 und dem Kondensator 154 bestehendes Siebglied.

Für die Beschreibung einer alternativen Form der Strombegrenzung wird auf DE 198 26 458.5 (intern: D215) verwiesen. Sie kann wie dort auch mit Hilfe eines Komparators aufgebaut werden und programmgesteuert sein.

Regelroutine

Fig. 20 zeigt eine mit Hilfe des μC 11 verwirklichte Regelung der Motordrehzahl.

In S404 wird die Regeldifferenz aus den Hallängen HL_s und HL_i (vgl. Fig. 16) berechnet. Die Berechnung wird in Fig. 21 näher erläutert. Das Ergebnis ist eine positive Stellgröße CNT_R für die Größe (Betrag) der Regeldifferenz und ein Vorzeichen VZ_R , das angibt, ob der Motor zu schnell ($VZ_R = 0$) oder zu langsam ($VZ_R = 1$) ist.

In S418 wird überprüft, ob $VZ_R = 1$ ist. Ist $VZ_R = 1$, so ist der Motor zu langsam, und der Kondensator 124 (Fig. 19) muß aufgeladen werden. Hierzu wird in S422 der Port RGL (Fig. 3, Fig. 19) durch $RGL := 1$ auf HIGH gesetzt. Analog wird in S420 der Port RGL durch $RGL := 0$ auf LOW gesetzt, falls $VZ_R = 0$ ist, der Motor also zu schnell ist.

Nach dem Setzen des Ports RGL wird beim Beispiel nach Fig. 20 eine zu dem Betrag CNT_R der Regeldifferenz proportionale Zeit gewartet, während der der Kondensator 124 und damit auch der Kondensator 127 (Fig. 19) aufgeladen bzw. entladen werden.

Hierzu wird in S424 eine Schleife begonnen. CNT_R wird um 1 vermindert, und es wird z. B. 10 μs lang gewartet. Daraufhin wird in S426 überprüft, ob $CNT_R > 0$ ist. Ist dies der Fall, so ist der als Schleifenzähler verwendete Wert CNT_R noch nicht abgearbeitet, und es wird zurück nach S424 gesprungen. Ist in S426 $CNT_R = 0$, so ist die Schleife insgesamt CNT_R mal durchlaufen worden, wobei mit CNT_R der in S404 berechnete ursprüngliche Wert CNT_R gemeint ist.

Nach der Beendigung der Schleife (S424 und S426) wird in S428 der Port RGL wieder auf TRISTATE gesetzt, und das Aufladen bzw. Entladen des Kondensators 124 ist beendet.

Fig. 21 zeigt an einem Zahlenbeispiel die Berechnung der Stellgröße CNT_R und des Vorzeichens VZ_R aus der Regeldifferenz, also S404 aus Fig. 20. In S440 wird die Differenz D_HL von der der Soll-drehzahl entsprechenden "Hallänge" HL_s und der gemessenen Hallänge HL_i (Fig. 16) berechnet. Als Beispiel wird eine Soll-drehzahl von 1000 U/min angenommen, welche einer "Hallänge" von 15000 μs und damit einer Zweibytendarstellung 0x3A98 entspricht. Die Ist-drehzahl beträgt hier 1100 U/min. was einer Hallänge von 13636 μs und einer Zweibytendarstellung 0x3544 entspricht. Die Differenz D_HL beträgt damit 0x0554 und ist in S440 auch binär dargestellt.

In S442 wird überprüft, ob die Differenz D_HL positiv ist. Ist dies der Fall, so wird in S444 VZ_R auf Null gesetzt, ansonsten wird in S446 D_HL aus der umgekehrten Differenz ($HL_i - HL_s$) berechnet und ist damit positiv, das Vorzeichen VZ_R wird auf 1 gesetzt.

In S448 wird D_HL drei mal nach rechts geshiftet, wobei in das höchstwertigste Bit (MSB) von HB immer eine Null

nachgeschoben wird. Das Shiften geschieht, da die Stellgröße CNT_R nur 1 Byte groß sein darf und die Regeldifferenz D_HL oft zu groß ist. Ein dreimaliges Shiften nach rechts entspricht einer ganzzahligen Division durch 8 ohne Berücksichtigung des Restes. Bei dem Shiften geht die Information der untersten drei Bits des LB verloren. Das Shiften des Beispiels aus S440 wird in Binärdarstellung gezeigt.

In S450 wird überprüft, ob nach dem dreimaligen Shiften nach rechts HB = 0 ist. Ist dies nicht der Fall, so ist das geschiftete D_HL immer noch größer als 1 Byte, und CNT_R wird in S458 auf den maximalen Wert 0xFF (binär: 11111111) gesetzt. Ist in S450 HB = 0, so wird in S452 überprüft, ob auch LB = 0 ist. Ist LB ungleich Null, so wird in S456 CNT_R auf LB gesetzt. In unserem obigen Beispiel wird CNT_R auf 0xAA gesetzt, dezimal entspricht dies 170. War in S452 LB = 0, so ist es möglich, daß trotzdem vor dem Shiften in S448 eine Regeldifferenz in den unteren drei Bits von LB bestanden hat. Um nicht auf Dauer eine kleine Regeldifferenz zu behalten, wird deshalb CNT_R in S454 auf 0x01 gesetzt. Das Vorzeichen VZ_R und der Wert CNT_R am Ende der Routine der Fig. 21 werden in Fig. 20 für die Regelung verwendet.

Fig. 22 zeigt das PWM-Signal OFF für den Fall, daß RGL = TRISTATE ist, Fig. 23 zeigt das Signal OFF beim Aufladen des Stellkondensators 124 und Fig. 24 das Signal OFF beim Entladen des Stellkondensators 124.

Fig. 22A, 23A und 24A zeigen die Schaltung der Regelung. Der Dreieckoszillator 35 ist an dem Minuseingang des Komparators 120 angeschlossen. Der Ausgang RGL ist mit einem Ersatzschaltbild im Inneren des µC 11 für den jeweiligen Zustand RGL = TRISTATE, RGL = HIGH und RGL = LOW dargestellt. Über den Strombegrenzungswiderstand 123 wird durch RGL der Stellkondensator 124 unverändert gelassen (Fig. 22), aufgeladen (Fig. 23) bzw. entladen (Fig. 24). Am Ausgang OFF liegt das PWM-Signal mit einem durch die Spannung U_C bestimmten Tastverhältnis vor.

Fig. 22B, 23B und 24B zeigen die zeitliche Veränderung der Spannung U_C bei dem jeweiligen Zustand des Ausgangs RGL, wobei in Fig. 23B und 24B RGL zum Zeitpunkt 194 von RGL = TRISTATE auf HIGH bzw. LOW umgeschaltet werden.

Fig. 22C, 23C und 24C zeigen jeweils in einem Spannungs-/Zeit-Diagramm die Spannung U_D des durch den Dreieckoszillator 35 erzeugten Dreiecksignals und die Spannung U_C des Stellkondensators 124 bei dem jeweiligen Zustand des Ausgangs RGL. Es ist zu beachten, daß hier der Augenblickswert der Dreiecksspannung U_D stets größer ist als der Wert 0.

Fig. 22D, 23D und 24D zeigen das aus Fig. 22C, 23C und 24C resultierende PWM-Signal OFF. In Fig. 22D bleibt das Tastverhältnis über die Zeit gleich, in Fig. 23D wird es bei zunehmendem U_C größer, und in Fig. 24C wird es bei abnehmendem U_C kleiner.

Fig. 23E zeigt einen vergrößerten Ausschnitt 192 aus Fig. 23B. Das Ansteigen der Spannung U_C des Kondensators wird immer wieder unterbrochen. Dies ist auf das in Fig. 29 beschriebene Aufteilen der Regelung in kleine zeitliche Portionen oder Häppchen zurückzuführen. In Fig. 29 wird bei jedem Aufruf der Regelung RGL auf HIGH gesetzt und der Kondensator 124 wird kurzzeitig aufgeladen. Vor dem Verlassen der Regelung wird RGL wieder auf TRISTATE gesetzt und die Spannung U_C am Kondensator 124 bleibt konstant. In Fig. 23E ist das Aufladen mit CH bezeichnet, und die konstanten Intervalle sind mit N-CH bezeichnet.

Fig. 24E zeigt analog einen vergrößerten Ausschnitt 193 aus Fig. 24B für den Fall, daß der Kondensator 124 entladen wird, ebenfalls für den Ablauf gemäß Fig. 29.

Fig. 25 zeigt den Gesamtablauf der Regelung. Die Tem-

peratur wird in S500 über den temperaturabhängigen NTC-Widerstand detektiert (Fig. 2 und 4). Über die A/D-Wandlung in S502 wird aus der Spannung am NTC-Widerstand ein digitaler Temperaturwert NTC_VAL (Fig. 2, 4, 5, 6 und 7). In S504 wird eine Hysterese der Kennlinie verwirklicht (Fig. 8 und 9). In einer Sensorabrißfunktion in S506 wird überprüft, ob ein Defekt des A/D-Wandlers vorliegt (Fig. 10). Bei einem Sensorabriß wird die Berechnung der Solldrehzahl umgangen, in S508 eine feste Sensorabrißdrehzahl n_SA gewählt und zu der Hall-Längenberechnung in S512 gesprungen. Mit dem NTC-Wert NTC_VAL aus der Hysteresefunktion wird in S510 über eine Kennlinienfunktion eine Solldrehzahl n_s berechnet (Fig. 11, 12, 13, 14 und 15). Die Solldrehzahl wird in S512 durch eine Hall-Längenberechnung in eine "Hall-Länge" HL_s umgerechnet. Die Ist-Drehzahl des Motors wird in S514 über einen Hall-Sensor detektiert, der ein Signal HALL liefert (Fig. 4, 16 und 17). Über die durch das Signal HALL ausgelösten Hall-Interrupts wird in S516 die Hall-Länge HL_i zwischen zwei Hall-Interrupts gemessen (Fig. 16 und 17). Mittels der beiden Hall-Längen für den Soll- und den Istwert berechnet der Regler in S518 einen Stellwert CNT_R für die Größe der Differenz von HL_s und HL_i und ein Vorzeichen VZ_R (Fig. 21). Anhand von CNT_R und VZ_R wird die Spannung U_C eines Kondensators 124 in S520 erhöht bzw. erniedrigt (Fig. 19 und 20). Über die Spannung U_C des Kondensators 124 wird in S522 das Tastverhältnis des PWM-Signals eingestellt (Fig. 19). Über das Tastverhältnis des PWM-Signals wird in S524 schließlich die Geschwindigkeit des Motors geregelt.

Funktionsmanager

Fig. 26 zeigt ein Flußdiagramm mit einer möglichen Ausführungsform des in dem µC 11 ablaufenden Gesamtprogramms. Nach dem Einschalten des Lüfters wird in dem µC 11 ein interner Reset ausgelöst. In S600 erfolgt die Initialisierung des µC 11. Es werden z. B. Parameter aus dem EEPROM 26 ausgelesen, der Kondensator 124 (Fig. 19) der PWM-Regelung wird auf einen Mindestwert aufgeladen und der Watchdog-Timer WDCNT 79 wird für die A/D-Wandlung gestartet.

Nach der Initialisierung erfolgt ein Sprung in einen sogenannten Funktionsmanager 190, der in S602 beginnt. Der Funktionsmanager regelt den Ablauf der einzelnen Unterprogramme.

Als erstes werden die Funktionen abgearbeitet, die zeitkritisch sind und bei jedem Durchlauf abgearbeitet werden müssen. Hierzu zählt die Kommunikationsfunktion COMM in S602, da der IIC-Bus 30 (Fig. 18) z. B. bei einer Baudrate von 2 k alle 250 µs überprüft werden muß. In S604 wird die A/D-Wandlung (S502, Fig. 25) aufgerufen und in S606 die Regelung RGL der Motordrehzahl (S518 und S520 in Fig. 25).

Fig. 27 zeigt ein beispielhaftes Funktionsregister 195, in dem für jede weitere Funktion ein Bit reserviert ist.

In diesem Beispiel ist das Funktionsregister 195 1 Byte groß, und es sind, von dem niederwertigsten Bit (LSB) beginnend, die folgenden Anforderungsbits für die unten erklärten anforderbaren Funktionen definiert:

- FCT_FILT für die Hysterese- und Sensorabrißfunktion,
- FCT_PT für die Kennlinienpunktbestimmungsfunktion,
- FCT_IPOL für die Drehzahlinterpolationsfunktion,
- FCT_HL für die Hall-Längenberechnungsfunktion für die Berechnung von HL_s.

Die restlichen Bits sind für weitere anforderbare Funktionen reserviert, die an den Funktionsmanager 190 angehängt werden können. Das Funktionsregister 195 belegt in diesem Ausführungsbeispiel 1 Byte, es kann aber um weitere Bytes erweitert werden.

Soll eine bestimmte anforderbare Funktion durch eine andere Funktion oder eine Interruptroutine angefordert werden, so wird das Bit der angeforderten Funktion auf 1 gesetzt. Das nächste Mal, wenn der Funktionsmanager 190 bei einem Durchlauf keine andere anforderbare Funktion mit höherer Priorität aufgerufen hat, so wird diese Funktion ausgeführt.

Ist eine angeforderte Funktion mit ihrer Abarbeitung fertig, so setzt sie ihr Bit (Fig. 27) wieder auf 0. Dies ermöglicht es, Funktionen, die nicht in einem Durchlauf abgearbeitet werden können, weil sie z. B. zu lange Zeit benötigen, aufzuteilen und in mehreren Aufrufen abzuarbeiten.

In Fig. 26 wird nach S606 in einer vorbestimmten Reihenfolge von der wichtigsten anforderbaren Funktion ausgehend jeweils geprüft, ob deren Anforderungsbit gesetzt ist. Ist dies bei einer Funktion der Fall, so wird diese ausgeführt, und daraufhin wird wieder an den Anfang S602 des Funktionsmanagers 190 gesprungen. Die Reihenfolge der Überprüfung des Funktionsregisters 195 gibt die Priorisierung der anforderbaren Funktionen vor. Je höher eine solche Funktion in dem Funktionsmanager 190 steht, desto höher ist ihre Priorität.

Die aufgerufenen Funktionen müssen dabei so kurz sein, daß ihre Abarbeitungszeit, addiert zu den immer ausgeführten Funktionen S602 bis S606, nie größer als die maximal zulässige Zeit zwischen zwei Abfragen des IIC-Buses ist. In dem obigen Beispiel mit einer Baudrate von 2 k und einer maximal zulässigen Zeit von 250 µs liegt die maximale Abarbeitungszeit für die in S610 bis S624 aufgerufenen Funktionen bei ca. 100 µs. Die in Fig. 25 aufgeführten Funktionen müssen also meistens in zeitlich kürzere Abschnitte untergliedert werden.

Die Untergliederung der Funktionen gemäß Fig. 26 stellt nur ein bevorzugtes Beispiel dar.

In S610 wird überprüft, ob das Anforderungsbit FCT_FILT für eine Filterfunktion gesetzt ist, also den Wert 1 hat. Ist es gesetzt, so wird nach S612 gesprungen, und die Hysteresefunktion (S504, Fig. 25) und die Sensorabriefunktion (S506 und S508, Fig. 25) werden ausgeführt. Diese werden als Filterfunktion bezeichnet, weil sie durch die Hysteresefunktion kleine Schwankungen des Wertes NTC_VAL in negative Richtung herausfiltern, und weil sie durch die Sensorabriefunktion unmögliche Werte NTC_VAL herausfiltern. Weiterhin sind Mittelwertbildungen (moving average) über vorhergehende Werte von NTC_VAL möglich.

War in S610 FCT_FILT nicht gesetzt, so wird in S614 überprüft, ob FCT_PT gesetzt ist. Ist dies der Fall, so werden je nach Berechnungsvariante einer oder beide den gemessenen Wert NTC_VAL umgebenden Kennliniendefinitionspunkte bestimmt und geladen (Teil von S510, Fig. 25).

Waren in S610 und S614 weder FCT_FILT noch FCT_PT gesetzt, und ist in S620 FCT_IPOL gesetzt, so wird in S622 die Berechnung des zum Wert NTC_VAL zugehörigen Soll-drehzahlwerts n_s ausgeführt (Teil von S510, Fig. 25).

Waren keine der in S610 bis S620 überprüften Bits gesetzt, und ist in S624 FCT_HL gesetzt, so wird aus der Soll-drehzahl n_s in S626 die "Hallänge" HL_s berechnet (S512, Fig. 25). Daraufhin wird wieder nach S602 gesprungen.

War in keiner der Abfragen bis S624 ein Anforderungsbit gesetzt, so wird ohne eine Aktion nach S602 zurückgesprungen, und die Funktionen, die bei jedem Durchlauf des Funk-

tionsmanagers 190 ausgeführt werden, werden erneut aufgerufen.

Durch den Funktionsmanager erhält man eine optimale Ausnutzung der Ressourcen des µC 11.

Im folgenden wird ein Überblick über das Zusammenarbeiten der verschiedenen Funktionen gegeben.

Die Kommunikationsfunktion in S602 wird in diesem Ausführungsbeispiel (Fig. 26) als erstes ausgeführt. Der externe IIC-Bus 30 (Fig. 18) wird abgelesen bzw. beschrieben, oder es werden Daten in das EEPROM geschrieben, bzw. aus diesem ausgelesen.

Die A/D-Wandlung in S604 wird ebenfalls bei jedem Durchlauf des Funktionsmanagers aufgerufen.

Fig. 28 zeigt die für den Funktionsmanager gemäß Fig. 26 modifizierte A/D-Wandlung aus Fig. 5. Der Initialisierungsteil 197 in S100-S104 wird im Initialisierungsteil INIT in S600 des Hauptprogramms ausgeführt. Der Sprung in die A/D-Wandlung erfolgt an Punkt A zwischen S106 und S108. Die A/D-Wandlung wird an Punkt B wieder verlassen, wenn also der Vergleich WDCNT > 0xFB das Ergebnis Nein (N) hat. Der Schritt S106 wird dabei nicht vergessen. Ist in S120 nicht mehr WDCNT > 0xFB, so wird wie in Fig. 5 in S122 der aus (CNT_HB 81, CNT_LB 82) gebildete Zähler CNT 80 inkrementiert, und daraufhin wird in S106 der Watchdog-Timer WDCNT wieder auf 0xFF gesetzt und der Watchdog-Timer von neuem gestartet.

Nach jeder abgeschlossenen A/D-Wandlung, wenn also in S118 ein neuer Wert NTC_VAL vorliegt, wird in S118 das Anforderungsbit FCT_FILT auf 1 gesetzt, damit bei der nächsten Möglichkeit in S612 (Fig. 26) die Filterfunktion ausgeführt wird.

In S606 (Fig. 26) wird die Regelung RGL (S518 und S520, Fig. 25) aufgerufen.

Fig. 29 zeigt eine beispielhafte Ausführung einer für den Funktionsmanager 190 angepaßte Regelung RGL. Es wird dabei von Fig. 20 ausgegangen. Gleiche oder gleich wirkende Teile wie in Fig. 20 werden deshalb mit denselben Bezugszeichen bezeichnet wie dort und gewöhnlich nicht nochmals beschrieben.

Die Berechnung von CNT_R und VZ_R in S404 benötigt so viel Zeit, daß die Regelung direkt nach der Berechnung verlassen wird. Dies wird durch das Einführen eines Flags FLAG_R erreicht. Ist in S400 nicht FLAG_R = 1, so liegt ein Neuaufruf der Regelung vor. In S402 wird mit FLAG_R := 1 das FLAG_R auf 1 gesetzt und so signalisiert, daß die Regelung aktiv ist. Daraufhin wird wie in Fig. 20 die Regelberechnung in S404 durchgeführt und aus der Regelung gesprungen.

Beim nächsten Aufruf der Regelung ist FLAG_R = 1, die Regelung ist also aktiv, und nun muß wie in den Schritten S418 bis S428 der Fig. 20 der Kondensator 124 aufgeladen bzw. entladen werden.

Das Aufladen bzw. Entladen des Kondensators 124 muß in kleinere Blöcke unterteilt werden, da das Laden bzw. Entladen des Kondensators 124 (Fig. 19) der PWM-Regelung oft länger als 100 µs benötigt.

Hierzu wird gemäß Fig. 29 die Regelung so verändert, daß der Kondensator 124 bei jedem Aufruf der Regelung (S606 in Fig. 26) häppchenweise auf- bzw. entladen wird, so daß kein Aufruf der Regelung länger als die durch den IIC-Bus maximal zulässige Zeit von z. B. 100 µs benötigt.

Hierzu werden vor der Entscheidung in S418 (Fig. 29) zwei weitere Variablen TMP_R und N und die Schritte S410 bis S416 eingeführt. In S410 wird der Variablen TMP_R der Wert (CNT_R-8) zugeordnet, also der Stellgrößenwert CNT_R, vermindert um die Zahl 8. Ist TMP_R > 0, so wird in S412 nach S416 gesprungen, ansonsten nach S414. In S416, wenn also CNT_R größer als 8 ist, wird einer Zähler-

variablen N der Wert 8 zugeordnet, und CNT_R wird der Wert TMP_R zugeordnet, so daß die neue Stellgröße CNT_R beim nächsten Aufruf der Regelung bekannt ist. In S414, wenn also CNT_R kleiner oder gleich 8 war, wird der Zählervariablen N der Wert CNT_R zugeordnet, und FLAG_R wird auf 0 gesetzt, da CNT_R bei diesem Aufruf der Regelung vollständig abgearbeitet wird.

In den Schritten S418 bis S428 wird wie in Fig. 20 abhängig vom Vorzeichen VZ_R der Kondensator 124 während des Durchlaufens einer Schleife aufgeladen bzw. entladen. Die Schleife wird jedoch im Gegensatz zu Fig. 20 nicht CNT_R mal durchlaufen, sondern N mal, also maximal 8 mal. Hierfür wurde in S424' und S426' CNT_R durch N ersetzt.

Es wird nun der Überblick über das Zusammenarbeiten der verschiedenen Funktionen aus Fig. 26 fortgesetzt.

Durch das Setzen von FCT_FILT durch die A/D-Wandlung in S604 (Fig. 26), wenn ein neuer NTC_VAL vorliegt, wird bei der nächsten Gelegenheit von S610 (Fig. 26) aus die Hysteresefunktion (Fig. 9) und die Sensorabrißfunktion (Fig. 10) aufgerufen werden.

Sind in S612 die Hysteresefunktion und die Sensorabrißfunktion ausgeführt, so wird FCT_FILT wieder auf 0 gesetzt, und in S612 wird FCT_PT auf 1 gesetzt, so daß bei dem nächsten Sprung nach S614 die Kennlinienbestimmungsfunktion S618 aufgerufen wird.

Ist in S618 die Kennlinienpunktbestimmungsfunktion fertig, so setzt sie FCT_PT auf 0, und FCT_IPOL auf 1. Bei dem nächsten Sprung nach S620 wird dann schließlich S622 aufgerufen. Ist die Drehzahlinterpolationsfunktion in S622 fertig, so setzt sie FCT_IPOL auf 0 und FCT_HL auf 1.

So wird dann in S624, da FCT_HL = 1 ist, die Hall-Längenberechnung S626 aufgerufen, und bei deren Beendigung wird FCT_HL wieder auf 0 gesetzt. In S626 wird ein Sollwert HL_s für die "Halllänge" berechnet.

Der Funktionsmanager erlaubt es, gegebenenfalls weitere Unterprogramme einzufügen und dabei die durch den IIC-Bus 30 gegebene Zeitbegrenzung einfach zu erfüllen. Weiterhin wird es den Unterprogrammen und den Interruptroutinen ermöglicht, andere Unterprogramme in einfacher Weise aufzurufen.

Der Aufbau des Hauptprogramms in der Form eines Funktionsmanagers eignet sich für alle Geräte, die einen Mikrocontroller oder Mikroprozessor aufweisen, der sowohl einen Bus als auch andere Aufgaben steuert, z. B. die Steuerung eines Kfz-Motors.

Es folgt eine Tabelle mit typischen Beispielen für die Werte der verwendeten Bauteile:

Kondensatoren:

135	1,5 nF
127, 152	10 nF
14, 90	22 nF
99, 110, 166, 167	33 nF
154	100 nF

Tantalkondensator:

124	3,3 µF
-----	--------

Widerstände:

140	3
-----	---

162, 163	47
94, 153, 155	1 k
133, 136	2,2 k
106	3,3 k
164, 165	4,7 k
123, 131, 132	10 k
170	22 k
92, 114, 126	33 k
134	47 k
16, 91, 93, 96, 101, 112, 128, 169	100 k

NTC-Widerstand:

18	NTC 100 k
----	-----------

npn-Transistor:

150	BC846
-----	-------

pnp-Transistoren:

95, 168	BC856B
---------	--------

Komparatoren:

108, 120, 130	LM2901 D
---------------	----------

Hall-Sensor:

40	HW101A
----	--------

EEPROM:

26	2-Wire Serial CMOS
----	--------------------

50

EEPROM

AT24C01A (ATM EL)

55

Mikrocontroller:

11	COP 842 CJ (Nat. Semicond.)
----	-----------------------------

Patentansprüche

65

1. Elektromotor (9) mit einer von einer variablen physikalischen Größe (2), insbesondere einer Temperatur, gesteuerten Drehzahl, mit einem in Form von digitalen Einzelwerten gespeicherten Kennlinienfeld zum Zu-

ordnen von Werten dieser physikalischen Größe zu entsprechenden Drehzahlwerten des Elektromotors, und mit einem Mikrocontroller (11) oder Mikroprozessor, der Zugriff zu den gespeicherten digitalen Einzelwerten hat.

2. Elektromotor (9) nach Anspruch 1, bei welchem die digitalen Einzelwerte mindestens teilweise in Vektorform gespeichert sind.
3. Elektromotor (9) nach Anspruch 1 oder 2, bei welchem dem Mikrocontroller (11) oder Mikroprozessor ein Programm zur Interpolation zwischen im Kennlinienfeld gespeicherten digitalen Einzelwerten zugeordnet ist.
4. Elektromotor (9) nach einem oder mehreren der vorhergehenden Ansprüche, mit einem A/D-Wandler (10) zum Umsetzen einer analogen physikalischen Größe (2) in einen Digitalwert.
5. Elektromotor (9) nach Anspruch 4, mit einer Anordnung zur Prüfung, ob bei der A/D-Wandlung ein Überlauf auftritt.
6. Elektromotor (9) nach Anspruch 5, bei welchem beim Auftreten eines Überlaufs der bei der A/D-Wandlung entstandene Digitalwert durch einen vorgegebenen Digitalwert ersetzt wird.
7. Elektromotor nach einem oder mehreren der vorhergehenden Ansprüche, bei welchem dem A/D-Wandler eine Hysteresefunktion (S504) nachgeschaltet ist, welche bei kleinen Änderungen der physikalischen Größe den bei einer vorhergehenden A/D-Wandlung ermittelten Digitalwert beibehält, um Drehzahlschwankungen des Motors zu reduzieren.
8. Elektromotor (9) nach Anspruch 7, bei welchem die Hysteresefunktion nur bei Änderungen der physikalischen Größe in einer Richtung wirksam ist, nicht aber bei Änderungen in der entgegengesetzten Richtung.
9. Elektromotor (9) nach einem oder mehreren der vorhergehenden Ansprüche, bei welchem dem A/D-Wandler (10) eine Plausibilitätsfunktion (S506) nachgeschaltet ist, welche den bei einer A/D-Wandlung ermittelten Digitalwert auf Plausibilität überprüft und einen nicht plausiblen Digitalwert durch einen vorgegebenen Digitalwert ersetzt.
10. Elektromotor (9) nach einem oder mehreren der vorhergehenden Ansprüche, bei welchem dem Elektromotor ein Drehzahlregler (6) zugeordnet ist, und das bei der A/D-Wandlung erzeugte Signal zur Erzeugung eines Sollwertsignals für diesen Drehzahlregler dient.
11. Elektromotor (9) nach Anspruch 10, bei welchem als Istwertsignal ein digitales Signal verwendet wird, welches im wesentlichen der Zeit proportional ist, die der Rotor des Elektromotors zum Durchlaufen eines vorgegebenen Drehwinkels benötigt.
12. Elektromotor (9) nach den Ansprüchen 10 und 11, bei welchem der bei der A/D-Wandlung ermittelte digitale Wert in einen Wert für die Zeit umgeformt wird (S512), die der Rotor (39) für das Durchlaufen des vorgegebenen Drehwinkels benötigen soll.
13. Elektromotor (9) nach Anspruch 12, bei welchem eine Differenz aus digitalem Sollwertsignal und digitalem Istwertsignal, oder umgekehrt, gebildet wird, um ein digitales Abweichungssignal (CNT_R) für das Maß der Regelabweichung und ein digitales Vorzeichensignal (VZ_R) für das Vorzeichen dieser Regelabweichung zu erhalten.
14. Elektromotor (9) nach Anspruch 13, bei welchem ein PWM-Steller für die Steuerung des Motorstroms vorgesehen ist, dessen Tastverhältnis von der Spannung an einem Kondensator (124) steuerbar ist, und

dieser Kondensator während einer Zeitdauer geladen oder entladen wird, die der absoluten Größe des digitalen Abweichungssignals im wesentlichen proportional ist, wobei durch das Vorzeichensignal gesteuert wird, ob der Kondensator geladen oder entladen wird.

15. Elektromotor (9) nach einem oder mehreren der vorhergehenden Ansprüche, bei welchem ein nichtflüchtiger Speicher (26) für die Speicherung der digitalen Werte des Kennlinienfelds vorgesehen ist.
16. Elektromotor (9) nach Anspruch 15, bei welchem mindestens ein Teil der im nichtflüchtigen Speicher (26) gespeicherten digitalen Einzelwerte veränderbar ist.
17. Elektromotor (9) nach Anspruch 16, welcher einen Anschluß (30) aufweist, der mit einem Eingabegerät für die Eingabe eines Werts in den nichtflüchtigen Speicher (26) verbindbar ist.
18. Elektromotor (9) nach Anspruch 16 oder 17, bei welchem eine Verbindung zu einem Datenbus vorgesehen ist, über den Daten in den nichtflüchtigen Speicher transferierbar sind.
19. Elektromotor (9) nach Anspruch 17 oder 18, bei welchem die Datenübertragung vom und/oder zum nichtflüchtigen Speicher (26) durch den Mikroprozessor oder Mikrocontroller (11) des Elektromotors steuerbar ist.
20. Elektromotor (9) nach Anspruch 19, bei welchem der Mikrocontroller (11) oder Mikroprozessor des Motors für die Datenverbindung mit einem äußeren Eingabegerät als sogenannter Slave dieser Datenverbindung ausgebildet ist.
21. Elektromotor (9) nach einem der vorhergehenden Ansprüche, bei dem die Betriebszeit gezählt wird.
22. Elektromotor (9) nach Anspruch 22, bei welchem die Betriebszeit von außen abgefragt werden kann.
23. Elektromotor (9) nach Anspruch 1 oder 2, bei welchem die digitalen Einzelwerte mindestens teilweise als Punkt mit zugehöriger Steigung gespeichert sind.
24. Verfahren zum Regeln einer physikalischen Größe, insbesondere einer Drehzahl, mit folgenden Schritten:
 - a) Aus einem gewünschten Wert für die physikalische Größe in digitaler Form und einem tatsächlichen Wert für die physikalische Größe, ebenfalls in digitaler Form, wird eine Differenz gebildet, um die Regelabweichung zu ermitteln;
 - b) Vorzeichen und Absolutwert dieser Differenz werden ermittelt;
 - c) abhängig vom Vorzeichen wird ein analoges Speicherglied geladen oder entladen;
 - d) die Dauer der Ladung oder Entladung erfolgt jeweils im wesentlichen proportional zur Größe des ermittelten Absolutwerts der Differenz;
 - e) mit einem von der Ladung des analogen Speicherglieds abhängigen Wert wird das Tastverhältnis eines Stellglieds beeinflusst, welches seinerseits mit seinem Ausgangssignal die zu regelnde physikalische Größe beeinflusst (Fig. 20).
25. Verfahren nach Anspruch 24, bei welchem der Vorgang der Ladungsänderung des analogen Speicherglieds in mindestens zwei zeitlich getrennte Vorgänge unterteilt wird, zwischen denen mindestens ein anderer Prozeß ausgeführt wird (Fig. 29).
26. Verfahren nach Anspruch 25, bei welchem der andere Prozeß ein Prozeß mit höherer Priorität ist.
27. Verfahren nach einem oder mehreren der Ansprüche 24 bis 26, bei welchem als Stellglied ein PWM-Steller verwendet wird, und, gesteuert von dem von der

Ladung des analogen Speicherglieds abhängigen Wert, das Tastverhältnis dieses PWM-Stellers beeinflusst wird (Fig. 22, 23 und 24).

28. Verfahren nach einem oder mehreren der Ansprüche 24 bis 27, bei welchem die als Regelabweichung ermittelte Differenz mit einem konstanten Faktor, insbesondere einem Faktor, der kleiner ist als 1, multipliziert wird (Fig. 21).

29. Verfahren nach Anspruch 28, bei welchem die Multiplikation durch Verschiebung einer als Binärwert vorliegenden Zahl in einem Register erfolgt (Fig. 21).

30. Verfahren nach einem oder mehreren der Ansprüche 24 bis 29, bei welchem die als Regelabweichung ermittelte Differenz bei Überschreiten einer vorgegebenen Größe durch einen vorgegebenen Wert ersetzt wird (Fig. 21).

31. Verfahren nach Anspruch 30, bei welchem ermittelt wird, ob bei der Bildung der Differenz ein Wert auftritt, welcher einen vorgegebenen maximalen Wert überschreitet, und in diesem Fall die Differenz durch den vorgegebenen Wert ersetzt wird (Fig. 21).

32. Verfahren nach Anspruch 28 oder 29, bei welchem dann, wenn der Multiplikationsvorgang den Wert Null für die Differenz ergeben hat, dieser Wert durch einen von Null abweichenden Wert ersetzt wird (Fig. 21).

33. Verfahren zum temperaturabhängigen Steuern der Drehzahl eines Elektromotors, mit folgenden Schritten:

a) In einem Speicher werden Wertcluster von Kennlinien-Definitionspunkten gespeichert, welche Wertcluster mindestens einen für eine bestimmte Temperatur kennzeichnenden Wert und eine dieser Temperatur zugeordnete Drehzahlinformation enthalten;

b) ein für die Motordrehzahl steuernde Temperatur kennzeichnender aktueller Wert wird in zeitlichen Abständen erfaßt;

c) dieser erfaßte Wert wird mit den gespeicherten, für die Temperatur kennzeichnenden Werten verglichen, welche in den gespeicherten Wertclustern enthalten sind;

d) ein dem aktuellen Wert benachbarter gespeicherter Wert wird ermittelt;

e) durch eine von diesem benachbarten Wert ausgehende Interpolation wird eine Drehzahlinformation zu dem erfaßten aktuellen Wert ermittelt;

f) ein von dieser interpolierten Drehzahlinformation abgeleiteter Wert (H_L) wird zur Vorgabe einer Soll-Drehzahl einem Drehzahlregler des Elektromotors zugeführt (Fig. 25).

34. Verfahren nach Anspruch 33, bei welchem die interpolierte Drehzahlinformation durch lineare Interpolation ermittelt wird (Fig. 14).

35. Verfahren nach Anspruch 34, bei welchem für mindestens einen Kennliniendefinitionspunkt zusätzlich zu dem für eine bestimmte Temperatur kennzeichnenden Wert und einer diesem Wert zugeordneten Drehzahlinformation eine Steigung für die lineare Interpolation abgespeichert wird (Fig. 12).

36. Verfahren nach einem oder mehreren der Ansprüche 33 bis 35, bei welchem die im Speicher gespeicherten Werte veränderbar sind (Fig. 18).

37. Verfahren nach einem oder mehreren der Ansprüche 33 bis 36, bei welchem der erfaßte für die Temperatur kennzeichnende Wert mit einem bei einer vorhergehenden Erfassung erfaßten, für die Temperatur kennzeichnenden Wert verglichen wird, und eine Änderung des zur Vorgabe der Soll-Drehzahl dienenden Werts nur vorgenommen wird, wenn entweder die Tempera-

tur zugenommen hat, oder wenn sie um einen vorgegebenen Mindestwert abgenommen hat (Fig. 9).

38. Verfahren nach Anspruch 37, bei welchem der vorgegebene Mindestwert als Variable gespeichert wird.

39. Verfahren nach Anspruch 38, bei welchem der vorgegebene Mindestwert über einen Bus im nichtflüchtigen Speicher veränderbar ist. (Fig. 18).

40. Verfahren nach einem oder mehreren der Ansprüche 33 bis 39,

bei welchem der erfaßte für die Temperatur kennzeichnende Wert auf Plausibilität geprüft wird, und ein vorgegebener Drehzahl-Sollwert erzeugt wird, wenn dieser erfaßte Wert einer in der Praxis nicht vorkommenden Temperatur, insbesondere einer zu tiefen Temperatur, entspricht (Fig. 10).

41. Verfahren nach Anspruch 40, bei welchem für die nicht vorkommende Temperatur eine Mindesttemperatur angenommen wird, und allen Temperaturen, die kleiner oder gleich dieser Temperatur sind, eine bestimmte Drehzahlinformation zugeordnet wird.

42. Verfahren nach Anspruch 41, bei welchem die vorgegebene Mindesttemperatur und die zugeordnete Drehzahlinformation in einem nichtflüchtigen Speicher gespeichert werden und veränderbar sind (Fig. 18).

43. Verfahren nach Anspruch 42, bei welchem der Elektromotor auf seine maximale Drehzahl eingestellt wird, wenn der erfaßte für die Temperatur kennzeichnende Wert einer in der Praxis nicht vorkommenden Temperatur entspricht.

44. Verfahren nach einem oder mehreren der Ansprüche 33 bis 43, bei welchem zur Regelung der Motordrehzahl eine Pulsweitenmodulation mit einem an einem Eingang eines Komparators anliegenden Dreieckssignal und einer an einem anderen Eingang des Komparators durch einen Kondensator vorgegebenen Spannung, welche durch einen Mikrocontroller veränderbar ist, verwendet wird (Fig. 19).

45. Verfahren nach Anspruch 44, bei welchem eine Änderung der Ladung des Kondensators mindestens einmal durch einen anderen Prozess unterbrochen wird (Fig. 29).

46. Verfahren zur A/D-Wandlung bei einer Anordnung mit einem einen temperaturabhängigen Widerstand enthaltenden Spannungsteiler, wobei ein Abgriff dieses Spannungsteilers das Potential an dem einen Eingang eines Komparators festlegt und das Potential an dem anderen Eingang des Komparators durch einen Kondensator bestimmt wird, der über eine Konstantstromquelle aufladbar ist, mit folgenden Schritten:

a) Zuerst wird der Kondensator entladen;

b) anschließend wird die Zeit gemessen, die der Kondensator bei Ladung durch die Konstantstromquelle benötigt, um das Potential des anderen Eingangs zu erreichen;

c) diese Zeit wird als Maß für die Temperatur des temperaturabhängigen Widerstands verwendet (Fig. 7).

47. Verfahren nach Anspruch 46, bei welchem ein Mikrocontroller oder Mikroprozessor vorgesehen ist, welcher im Schritt a) die Entladung des Kondensators bewirkt und welcher zur Zeitmessung gemäß Schritt c) dient (Fig. 4).

48. Elektromotor (9) mit in einem Speicher abgespeicherten diskreten Werten, welche in Form von Stützwerten eine Temperatur/Drehzahl-Kennlinie definieren, wobei die diskreten Werte über eine Datenverbindung zu einem außerhalb des Elektromotors angeordneten Eingabegerät veränderbar sind.

49. Elektromotor (9) nach Anspruch 48, bei welchem der Speicher ein nichtflüchtiger Speicher (26) ist.
50. Motor (9) nach Anspruch 49, bei welchem der Speicher ein EEPROM (26) ist.
51. Motor (9) nach einem der Ansprüche 48 bis 50, bei welchem ein Mikroprozessor oder Mikrocontroller mit einer Verbindung zu einem externen Bus versehen und dazu ausgebildet ist, die diskreten Werte von dem externen Bus an den nichtflüchtigen Speicher weiterzugeben.
52. Motor nach Anspruch 51, bei welchem dem Mikrocontroller (11) eine Paßwortfunktion für einen Zugriff über den externen Bus zugeordnet ist.
53. Motor nach Anspruch 51 oder 52, bei welchem der Mikrocontroller eine Checksummenfunktion für im Speicher abgespeicherte diskrete Werte aufweist.
54. Verfahren zum Betrieb eines elektronisch kommutierten Motors, welchem ein Mikroprozessor oder Mikrocontroller und ein diesem zugeordnetes Programm zugeordnet sind, welches Programm zur Steuerung einer Mehrzahl von Motorfunktionen unterschiedlicher Priorität dient, mit folgenden Schritten:
- a) Es wird eine Mehrzahl von für den Betrieb des Motors erforderlichen, anforderbaren Routinen vorgesehen;
 - b) wenn eine anforderbare Routine benötigt wird, wird ein entsprechendes Anforderungssignal für sie gesetzt;
 - c) durch eine übergeordnete Programmfunktion wird geprüft, welche angeforderte Routine die höchste Priorität aufweist, und diese Routine höchster Priorität wird als erste ausgeführt;
 - d) nach Ausführung dieser Routine höchster Priorität wird das dieser Routine zugeordnete Anforderungssignal rückgesetzt (Fig. 26).
55. Verfahren nach Anspruch 54, welches nach Art einer Schleife im Betrieb des Motors ständig wiederholt wird (Fig. 26).
56. Verfahren nach Anspruch 54 oder 55, bei welchem eine im Programm auszuführende anforderbare Routine während ihrer Ausführung ein Anforderungssignal für eine andere auszuführende anforderbare Routine erzeugt.
57. Verfahren nach einem der Ansprüche 54 bis 56, bei welchem die übergeordnete Programmfunktion jeweils nach Abarbeitung einer angeforderten Routine eine für den Betrieb des Motors erforderliche Folge von zeitkritischen Programmschritten durchläuft, bevor sie die nächste angeforderte Routine startet (Fig. 26).
58. Verfahren nach Anspruch 57, bei welchem die übergeordnete Programmfunktion nach Abarbeitung einer anforderbaren Routine jeweils eine am Motor vorgesehene Verbindung zu einem Datenbus abfragt (Fig. 26).
59. Verfahren nach Anspruch 58, bei welchem die maximale Zeitdauer der anforderbaren Routinen durch die gewünschte Baudzahl der Datenübertragung über den Datenbus festgelegt wird (Fig. 26).
60. Verfahren nach einem oder mehreren der Ansprüche 54 bis 59, bei welchem die Kommutierung des Motors jeweils während eines Interrupts erfolgt, der den augenblicklichen Programmablauf unterbricht (Fig. 17).
61. Verfahren nach Anspruch 60, bei welchem der gerade in Ausführung befindliche Befehl vor der durch den Interrupt erfolgenden Unterbrechung des augenblicklichen Programmablaufs abgearbeitet wird.
62. Verfahren nach einem oder mehreren der Ansprü-

- che 60 bis 61 für einen elektronisch kommutierten Motor, welcher zur Erfassung seiner Rotorstellung einen Hallgenerator aufweist, wobei ein Interrupt durch eine vorgegebene Art einer Änderung des Ausgangssignals dieses Hallgenerators ausgelöst wird.
63. Verfahren nach einem oder mehreren der Ansprüche 54 bis 62, bei welchem die angeforderte Funktion ihr Anforderungssignal zurücksetzt, sobald sie vollständig abgearbeitet ist (Fig. 26).
64. Motor (9) mit einem Mikroprozessor oder Mikrocontroller (11) und einem Bus, bei welchem der Mikroprozessor oder Mikrocontroller (11) sowohl den Bus als auch den Motor steuert.
65. Motor (9) nach Anspruch 64, bei welchem der Mikroprozessor oder Mikrocontroller (11) zusätzlich die A/D-Wandlung eines dem Motor zugeführten analogen Signals steuert.
66. Motor (9) nach Anspruch 64 oder 65, bei welchem der Mikroprozessor oder Mikrocontroller (11) zusätzlich eine Temperatur/Drehzahl-Kennlinie verwirklicht.
67. Motor (9) nach einem oder mehreren der Ansprüche 64 bis 66, bei welchem der Mikroprozessor oder Mikrocontroller (11) einen Paßwortschutz für den Zugriff auf mindestens eine Funktion über den Bus aufweist.
68. Motor (9) nach einem oder mehreren der Ansprüche 64 bis 67, bei welchem der Mikroprozessor oder Mikrocontroller (11) einen Betriebsstundenzähler aufweist.

Hierzu 29 Seite(n) Zeichnungen

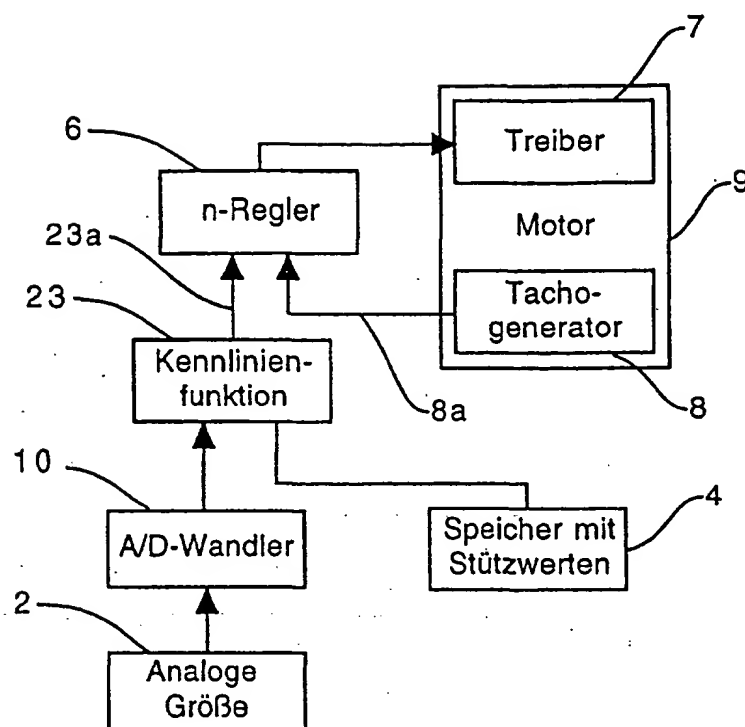


Fig. 1

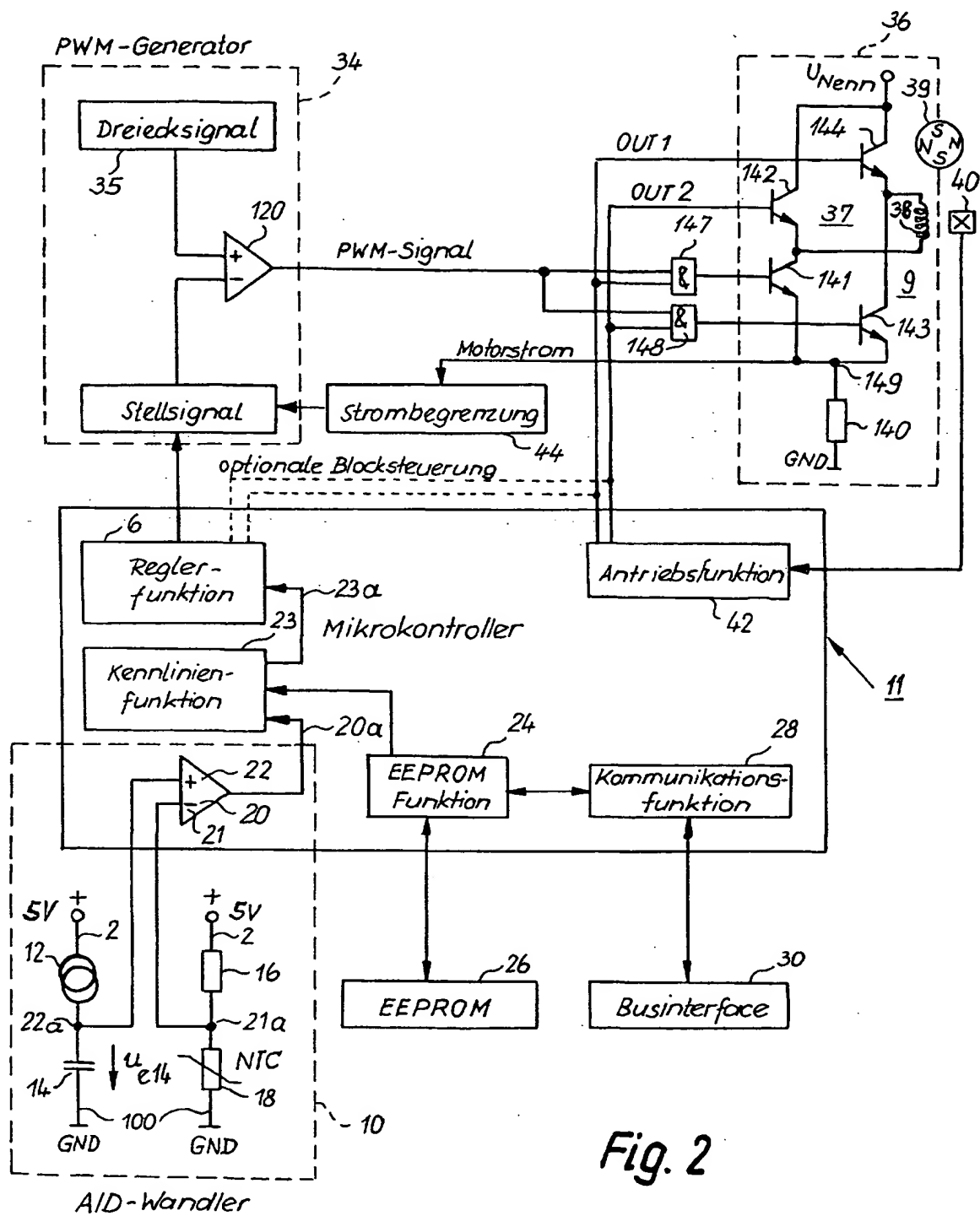


Fig. 2

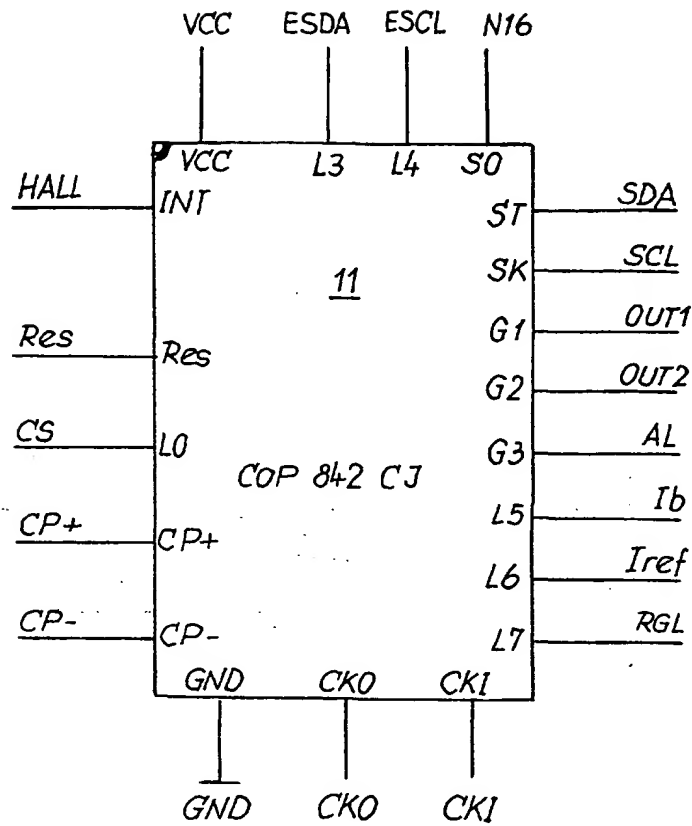


Fig. 3

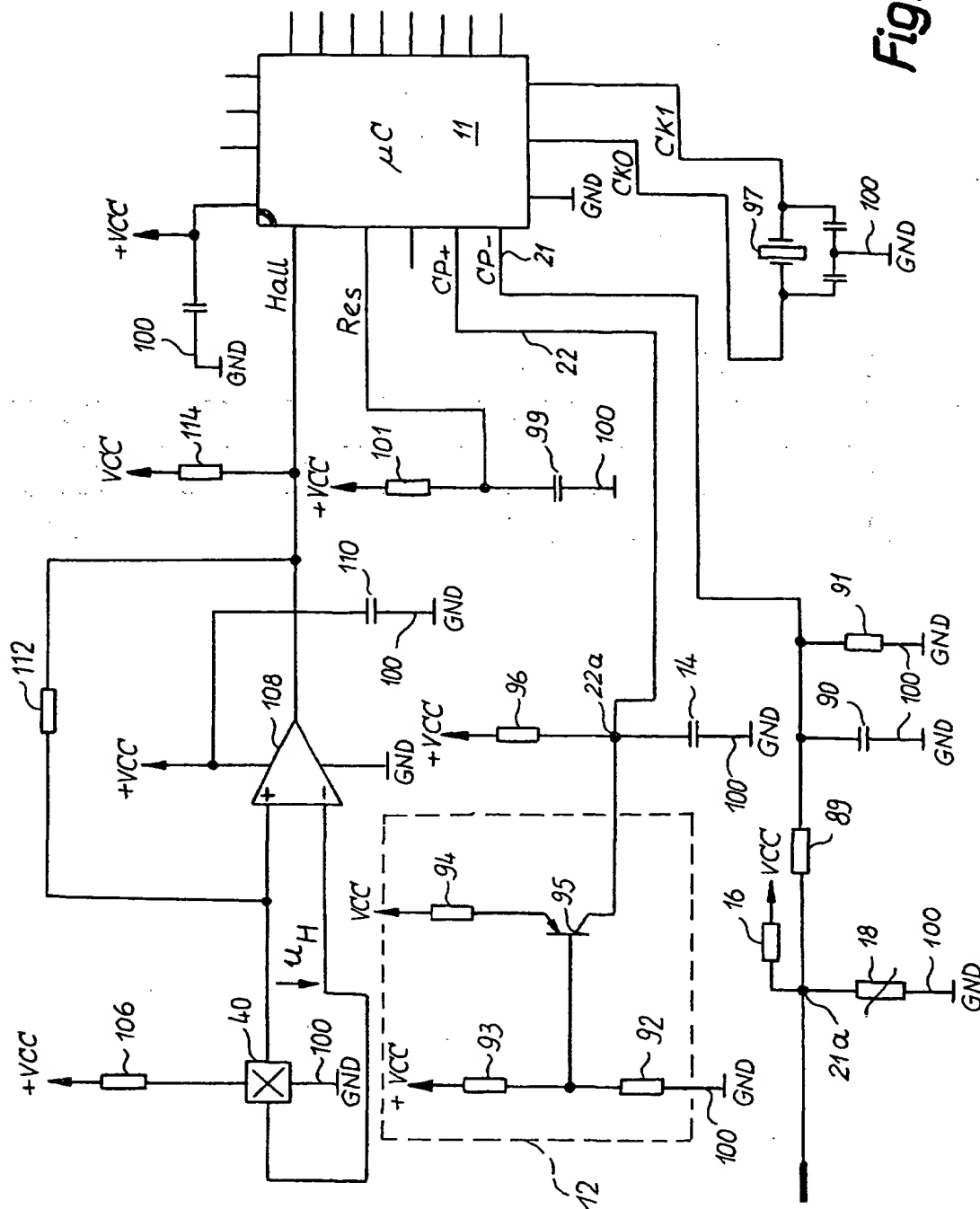


Fig. 4

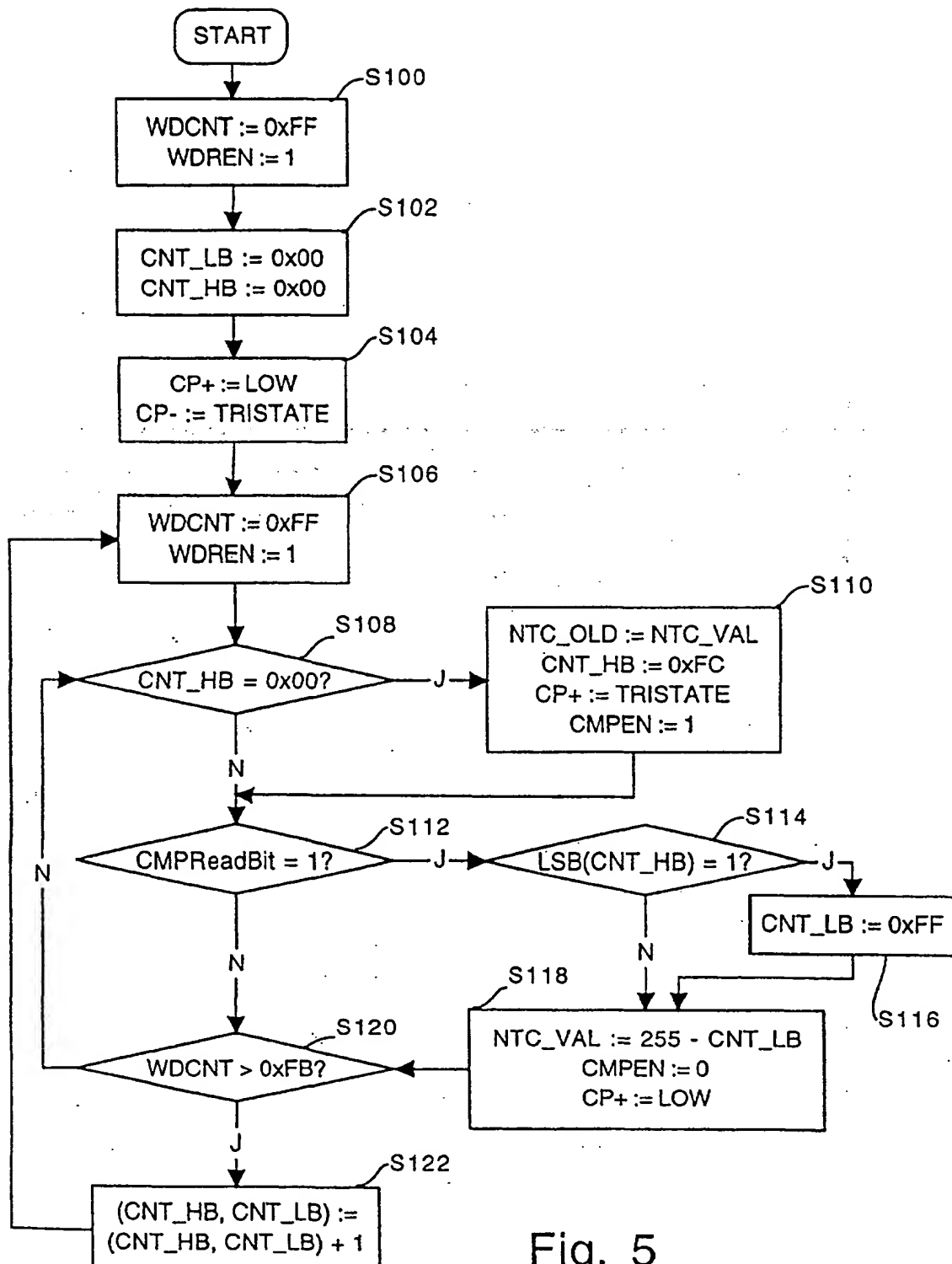


Fig. 5

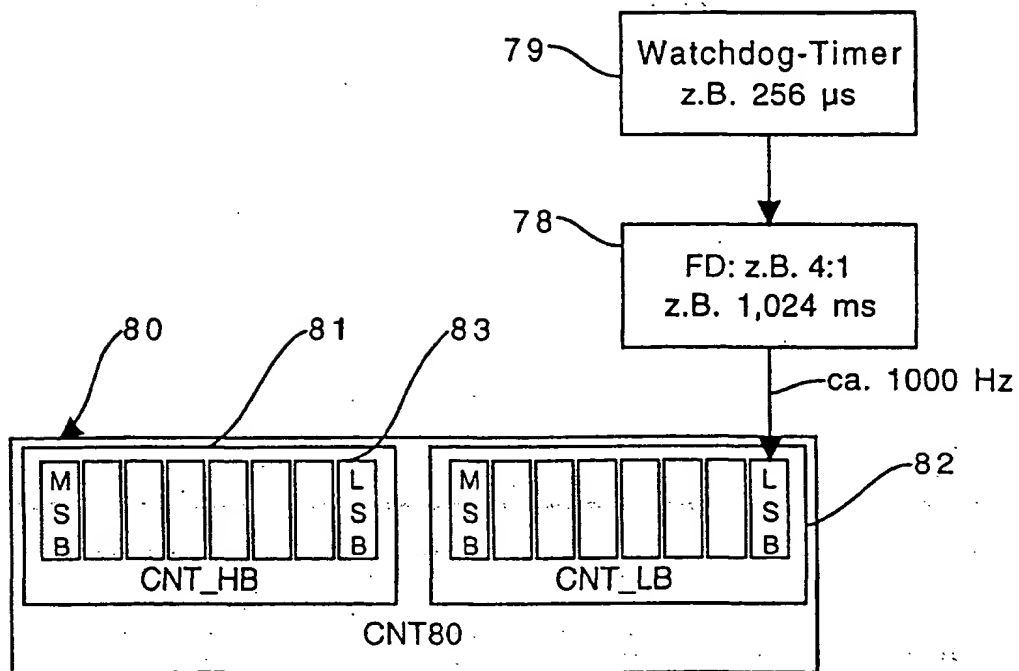


Fig. 6

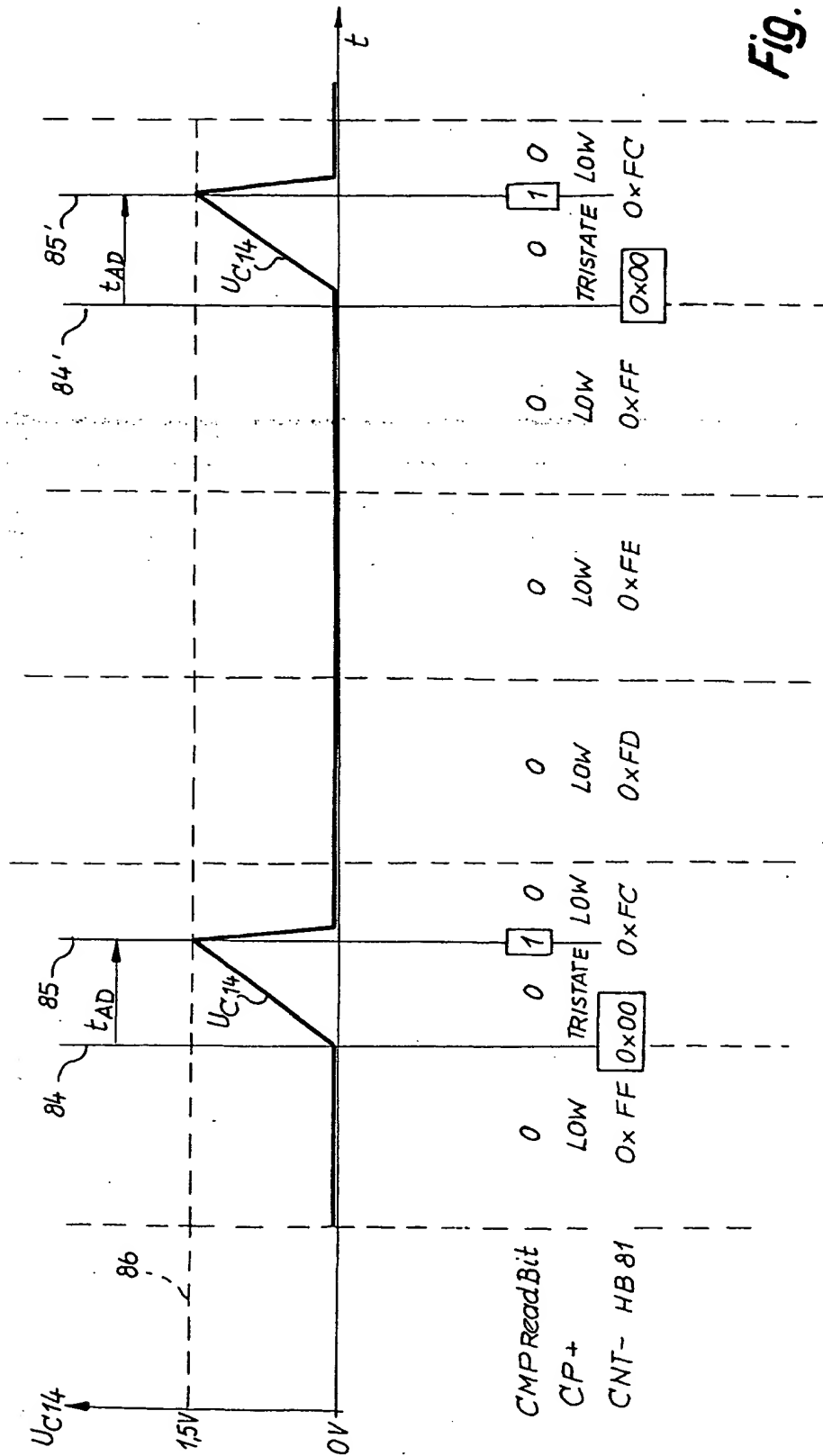


Fig. 7

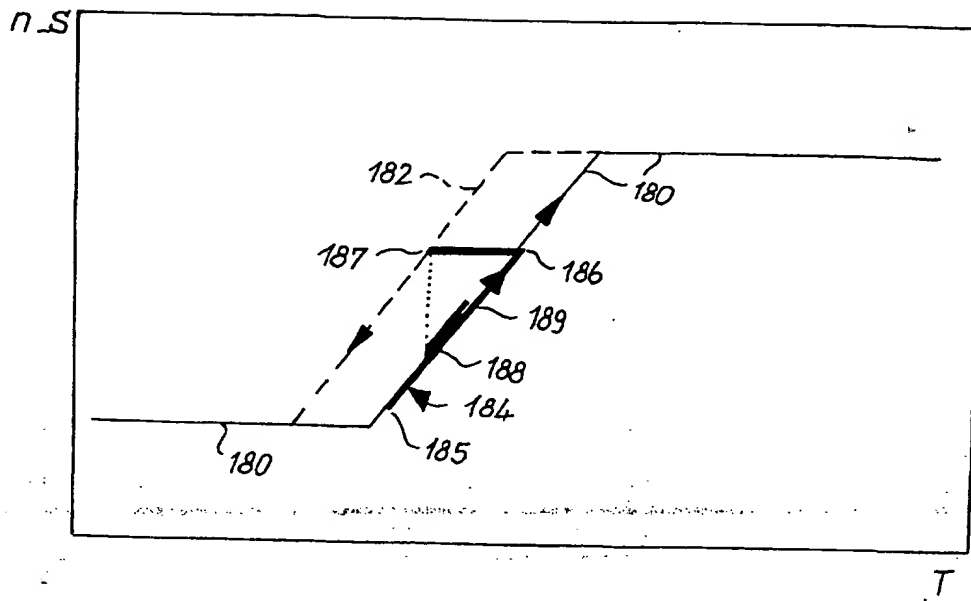


Fig. 8

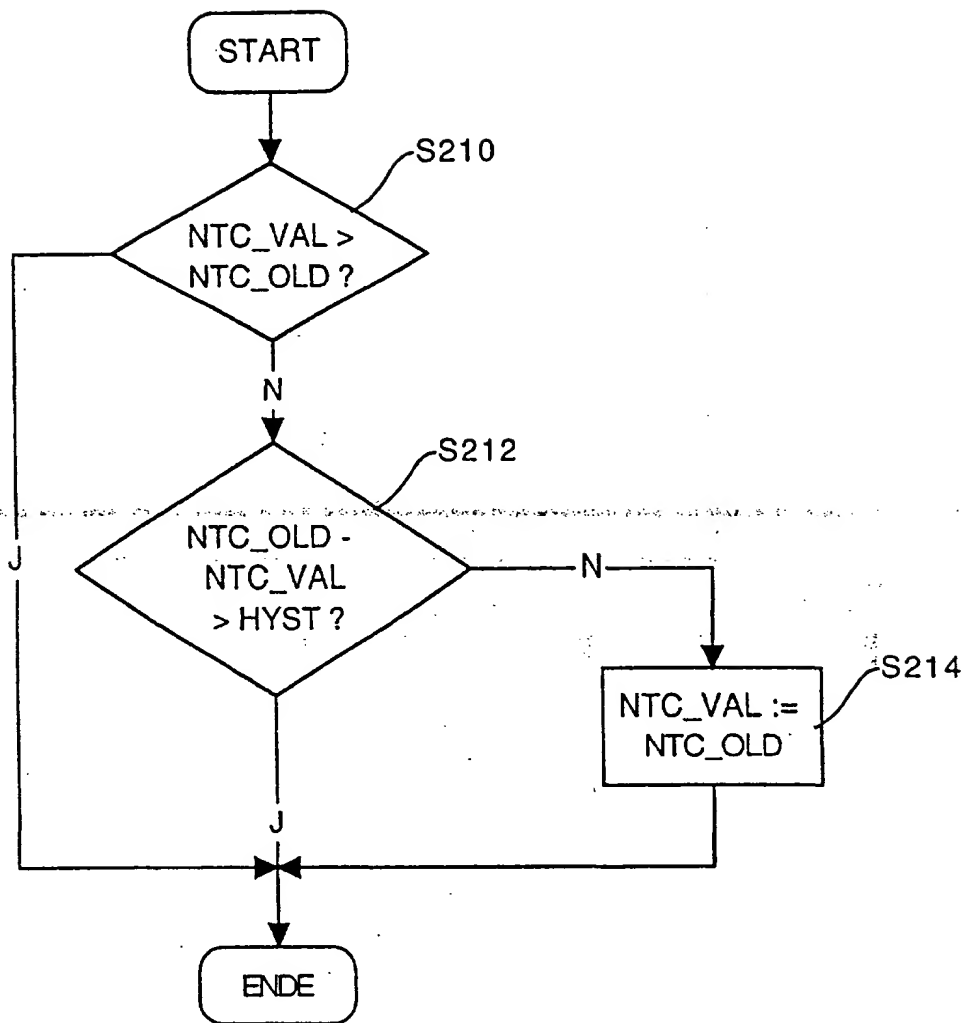


Fig. 9

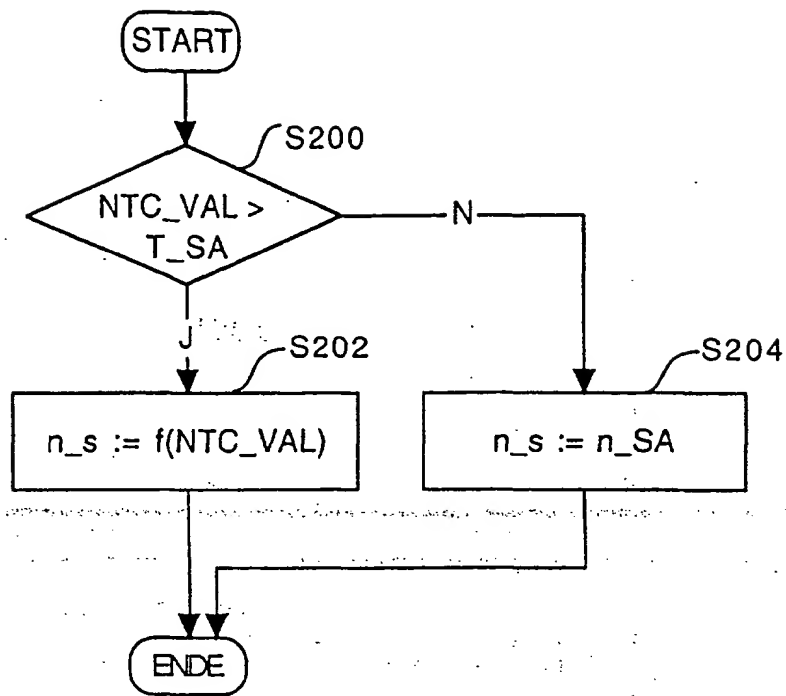


Fig. 10

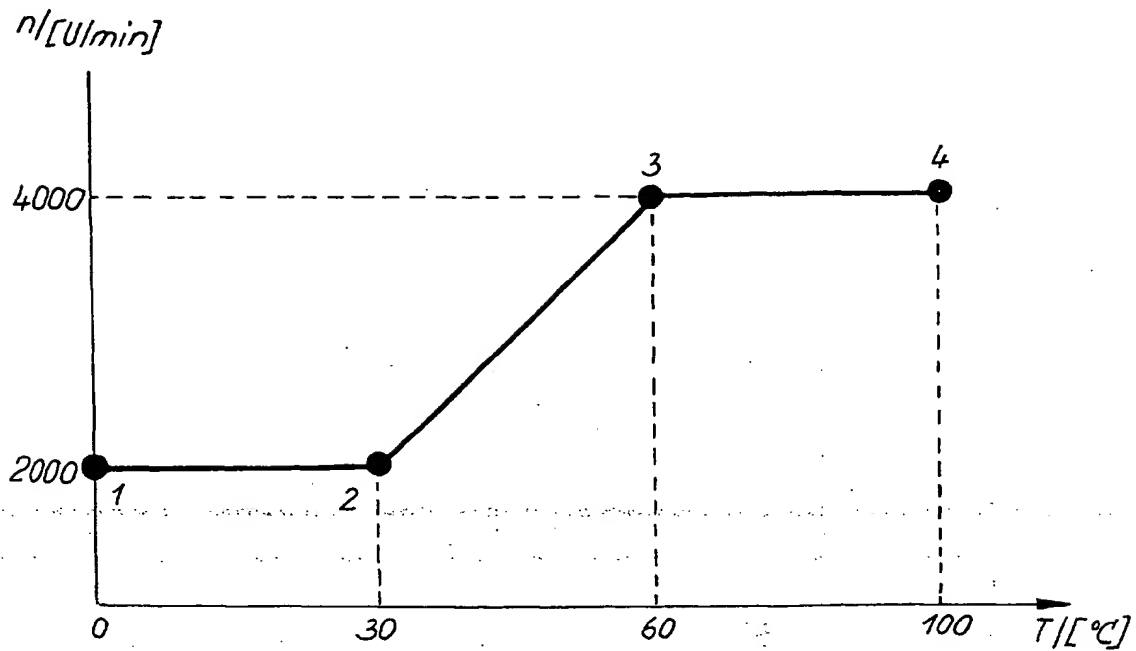


Fig. 11

<i>p</i>	<i>T</i>		<i>n</i>		<i>S</i>
	$^{\circ}\text{C}$	<i>hex</i>	min^{-1}	<i>hex</i>	<i>hex</i>
1	0	0x39	2000	0x07D0	0x0000
2	30	0x8D	2000	0x07D0	0x001E
3	60	0xCF	4000	0x0FA0	0x0000
4	100	0xF1	4000	0x0FA0	0x0000

Fig. 12

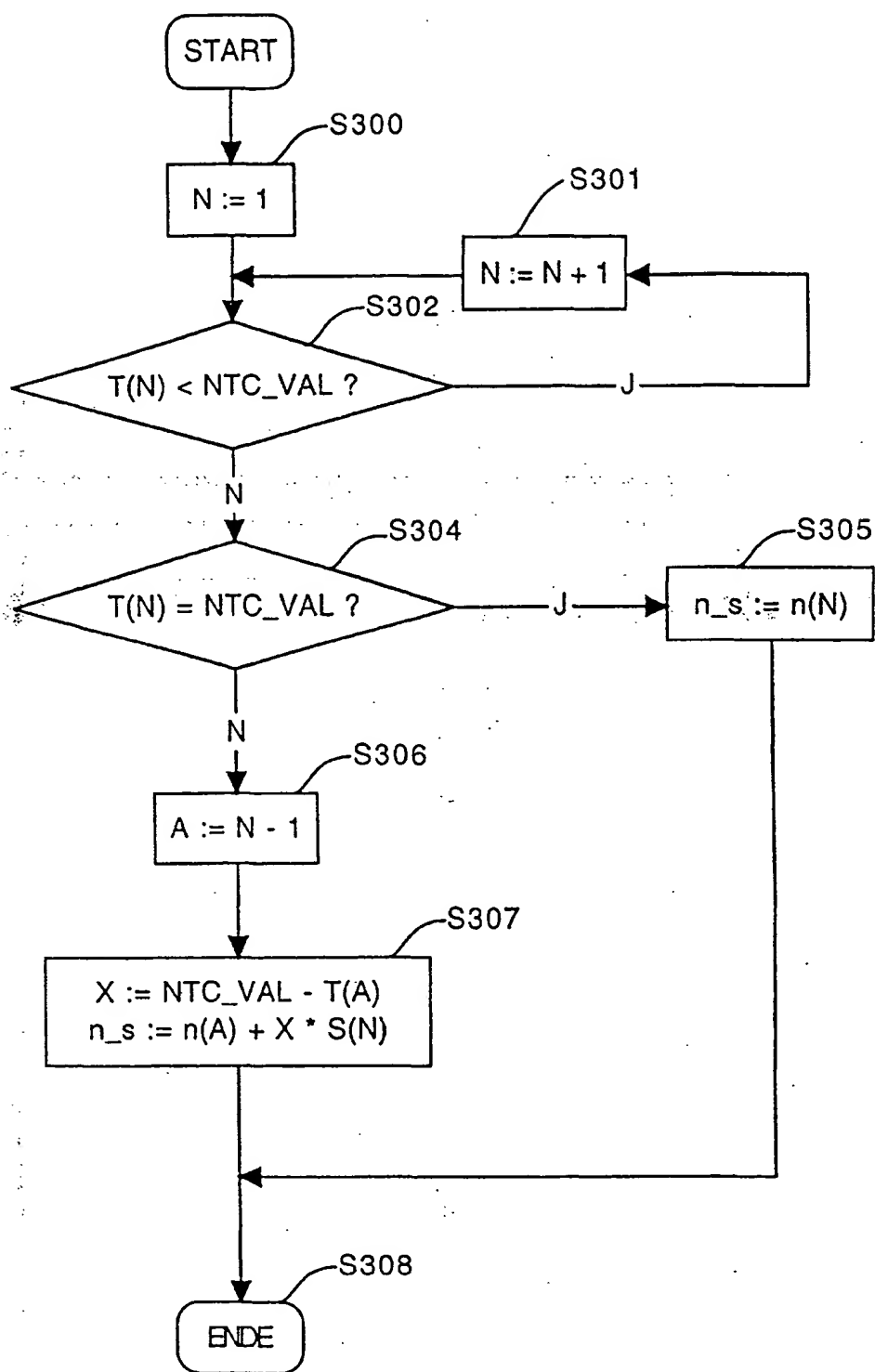


Fig. 13

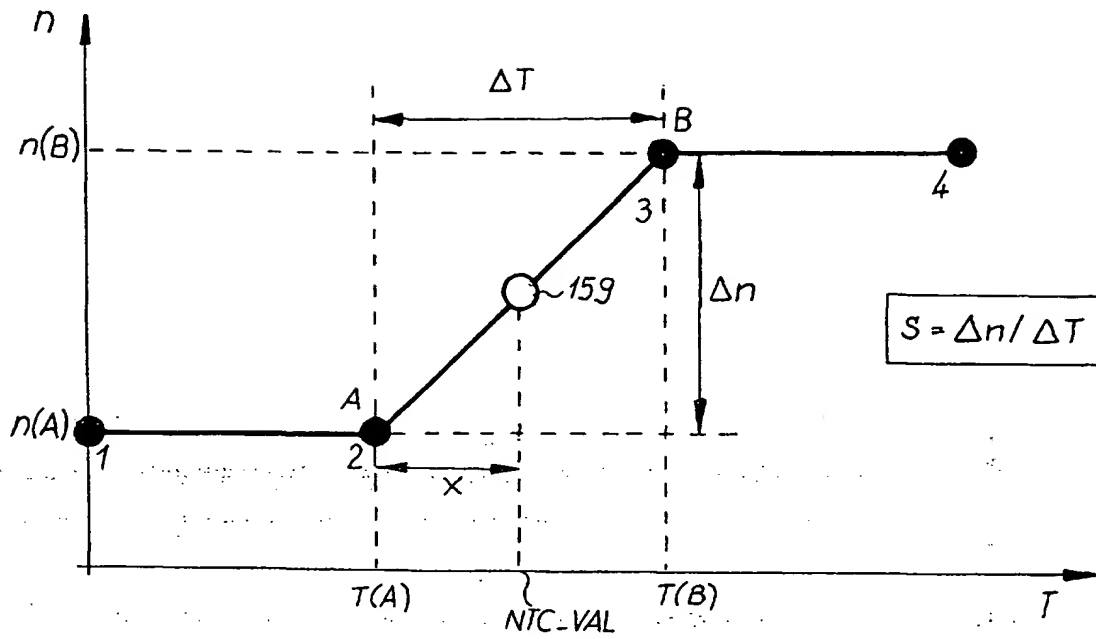


Fig. 14

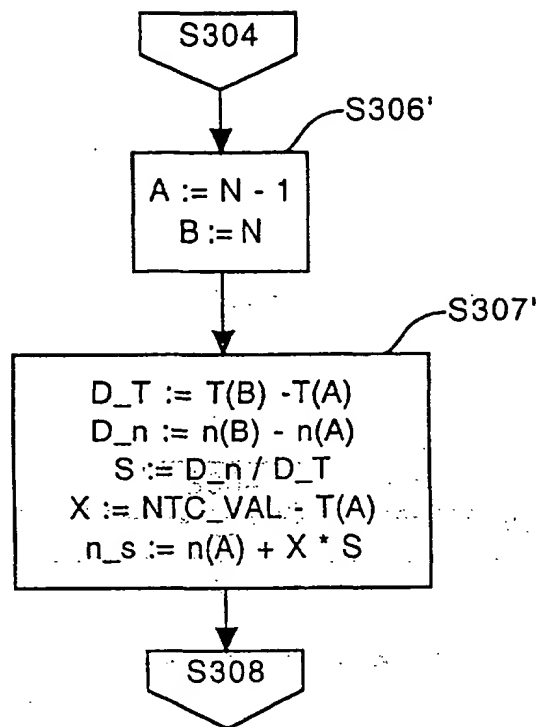


Fig. 15

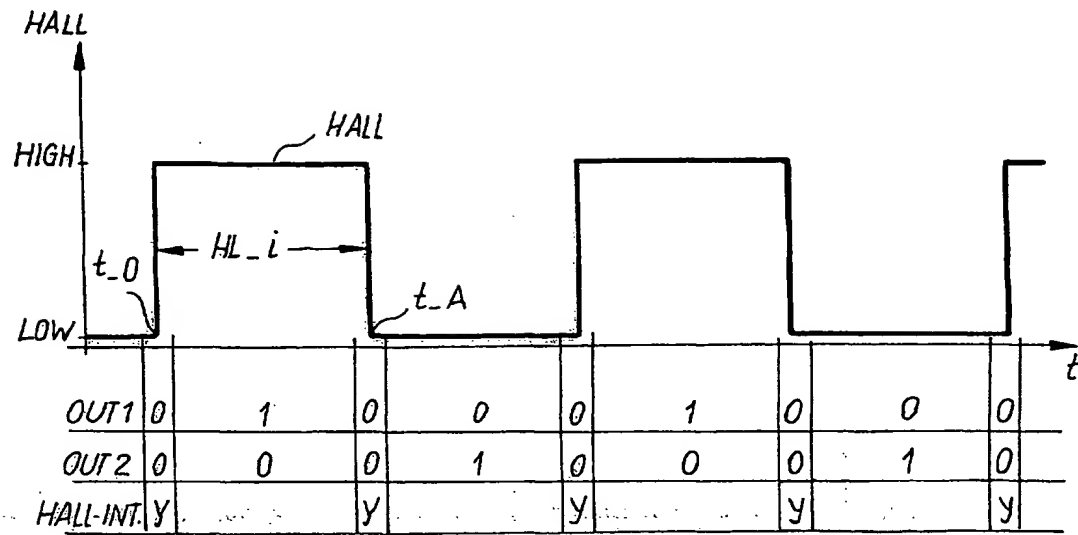


Fig. 16

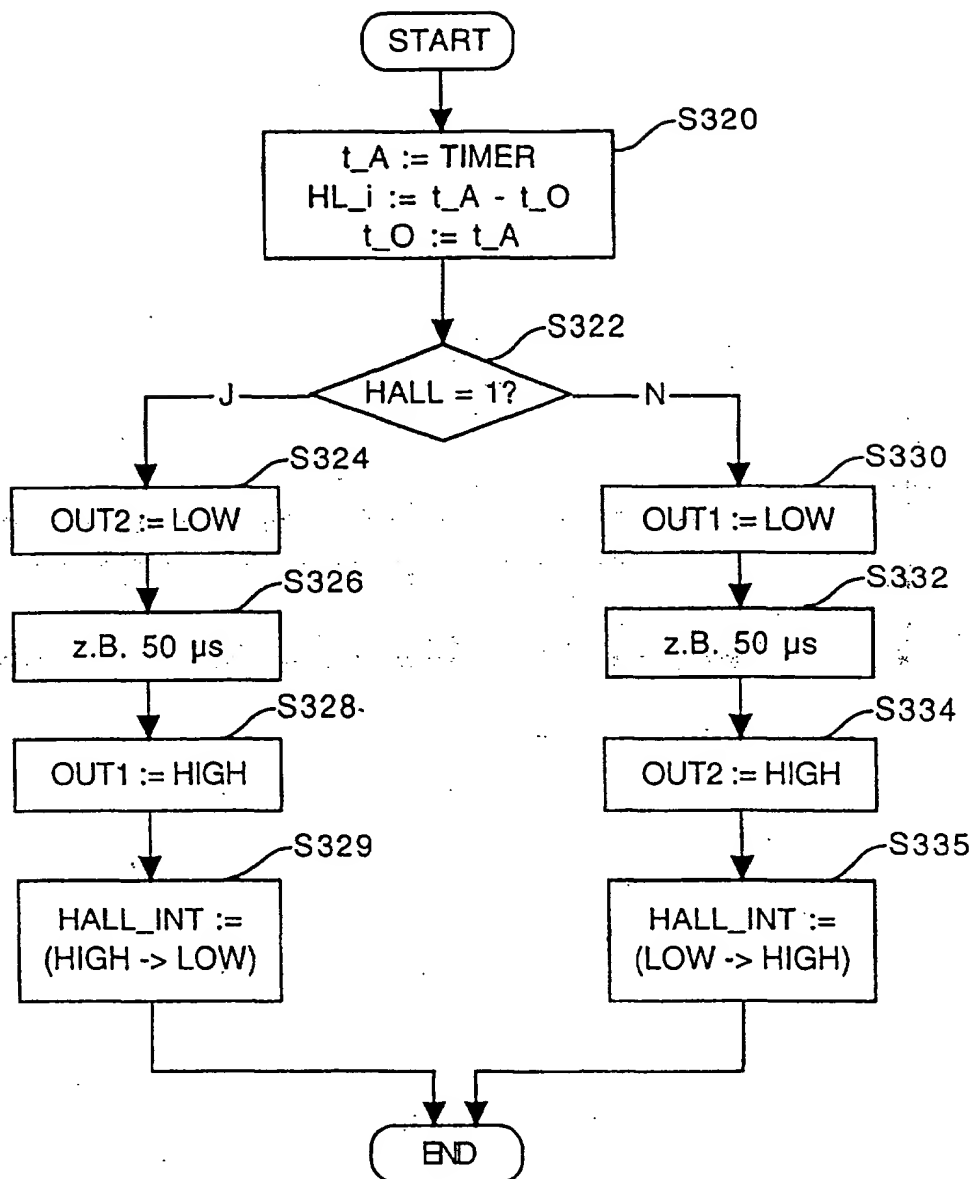


Fig. 17

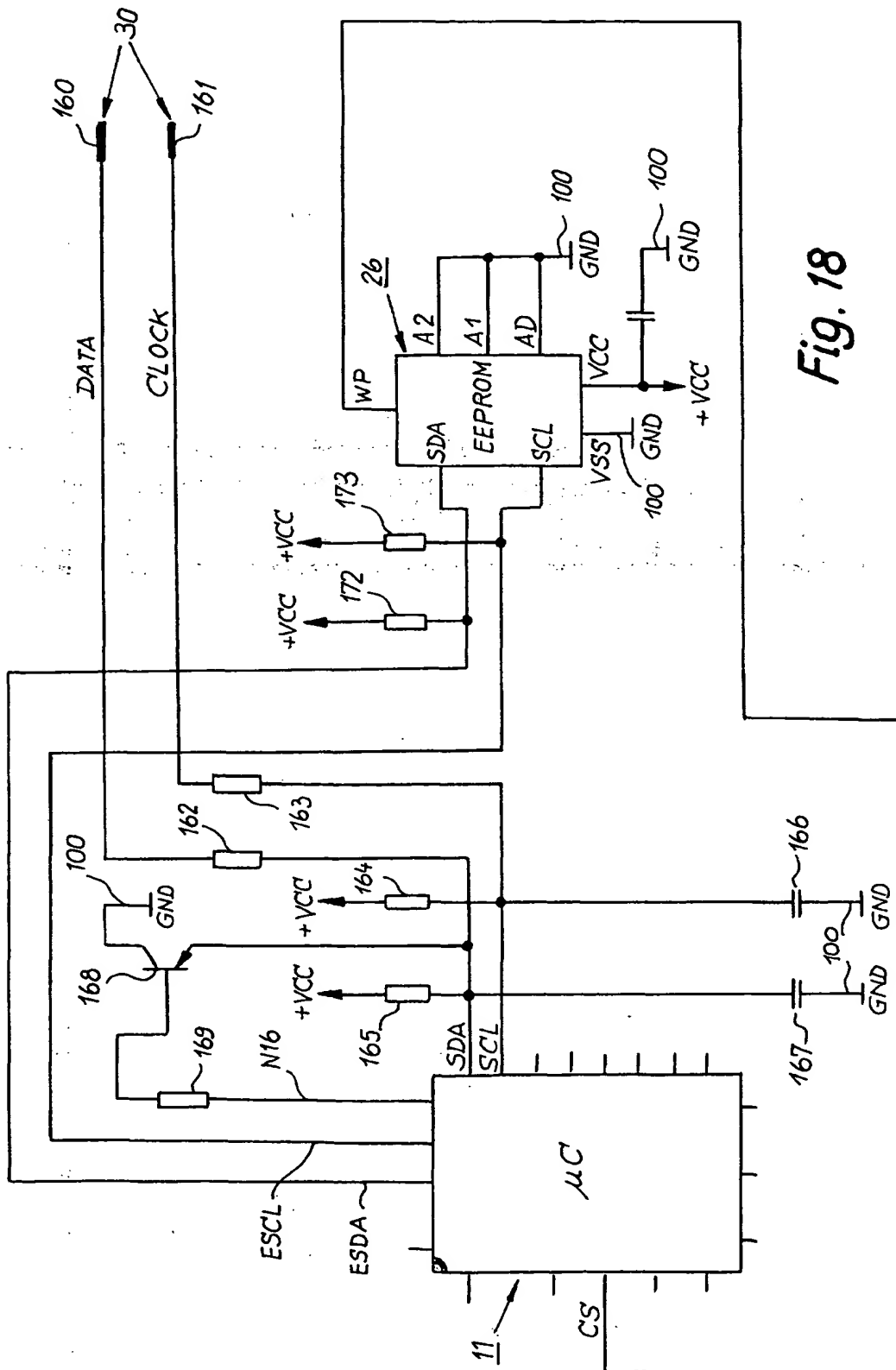


Fig. 18

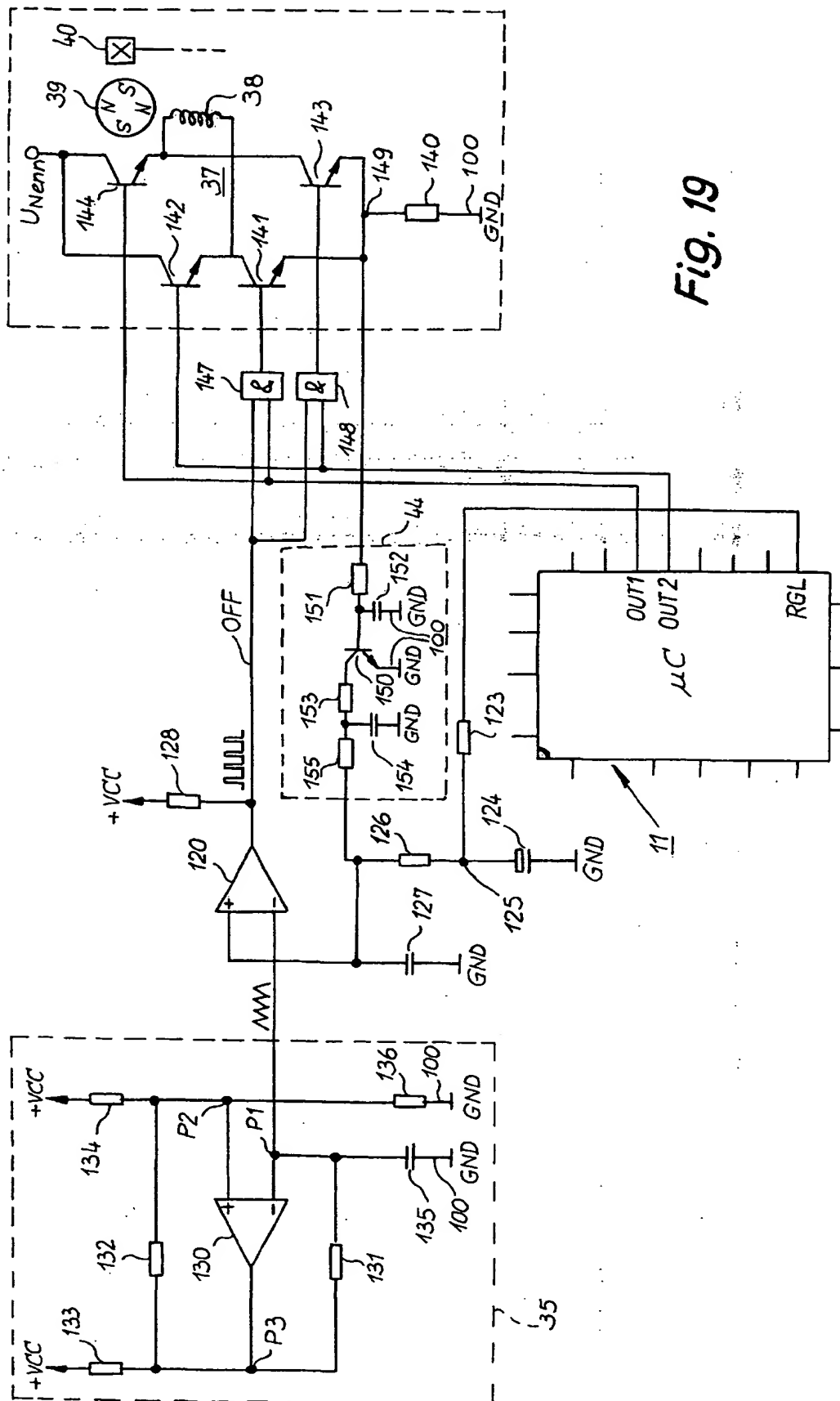


Fig. 19

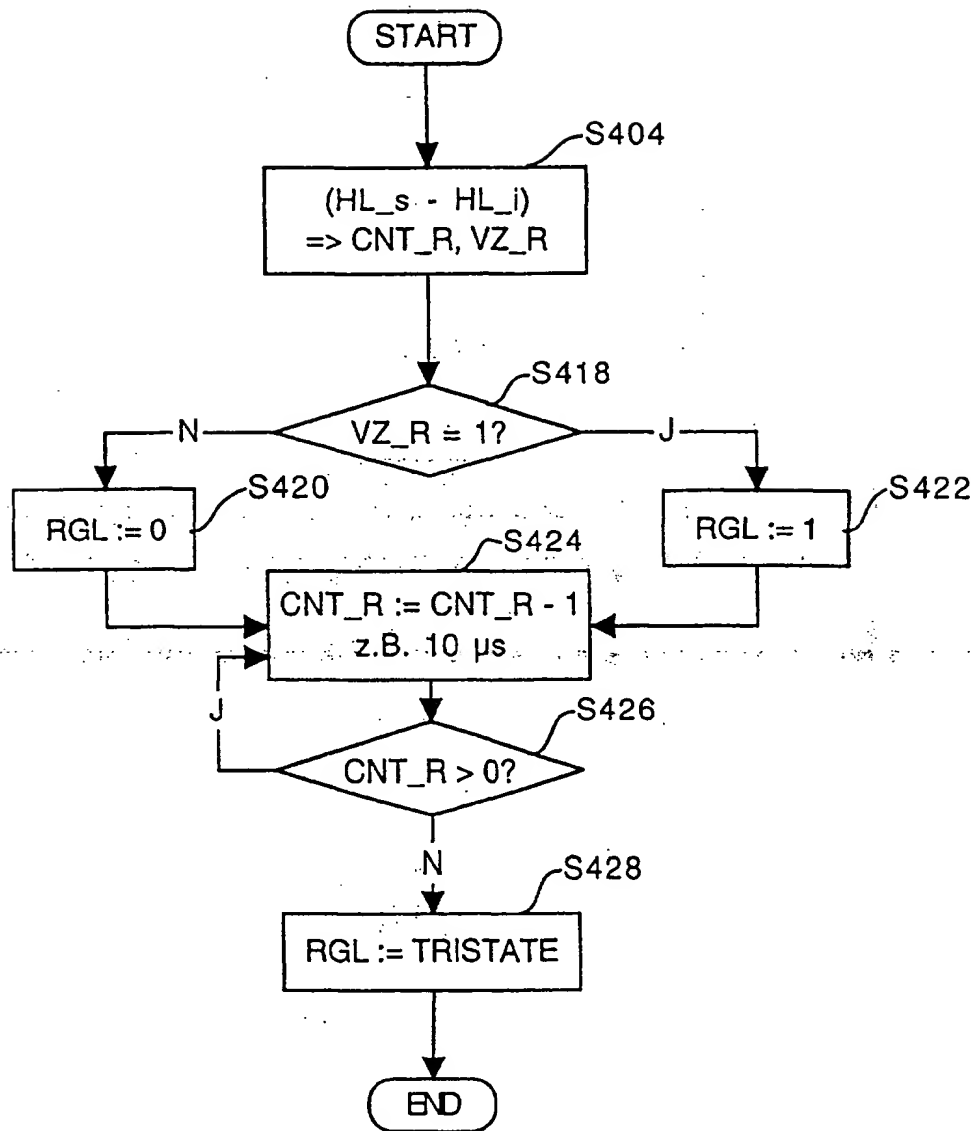


Fig. 20

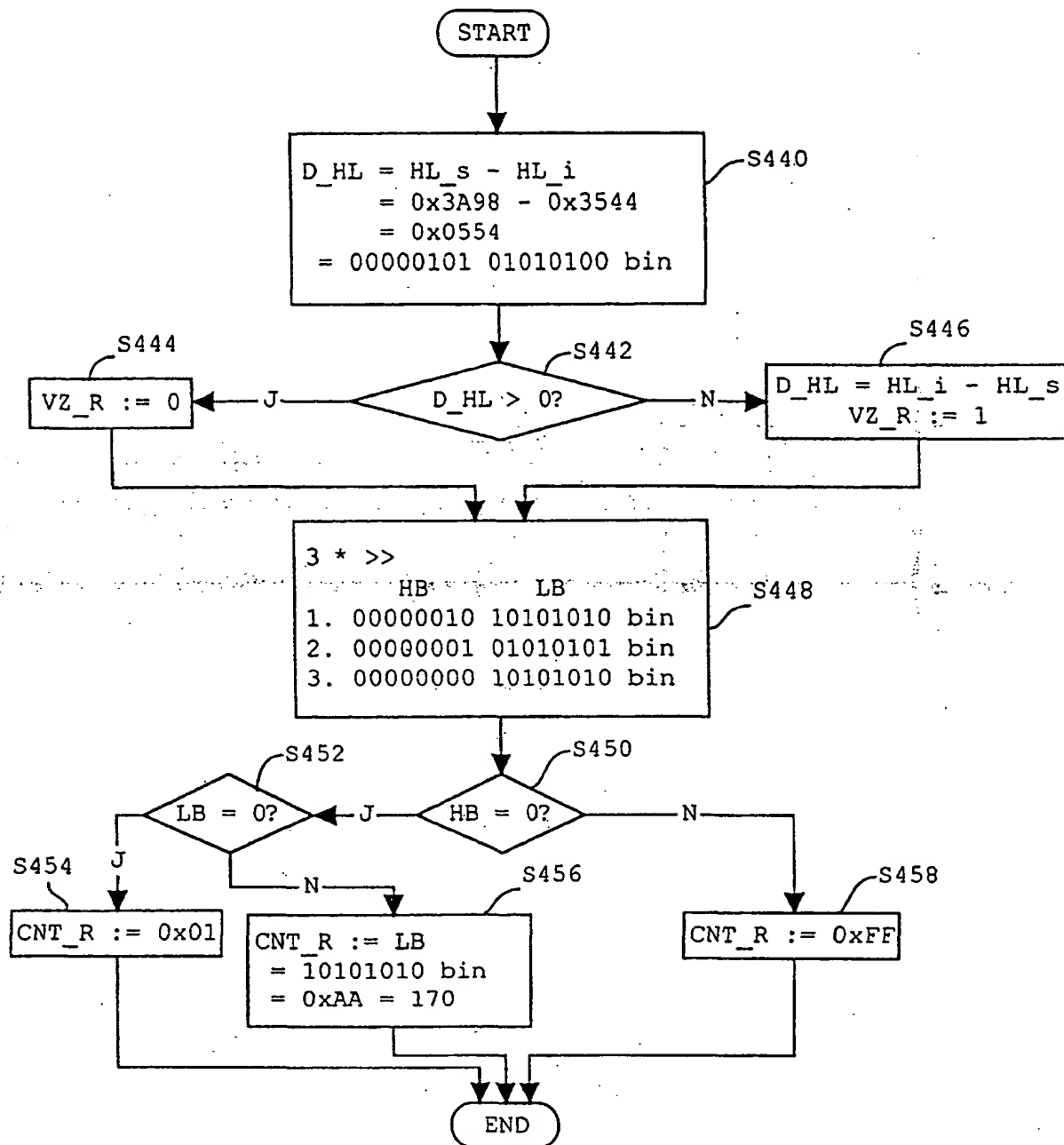


Fig. 21

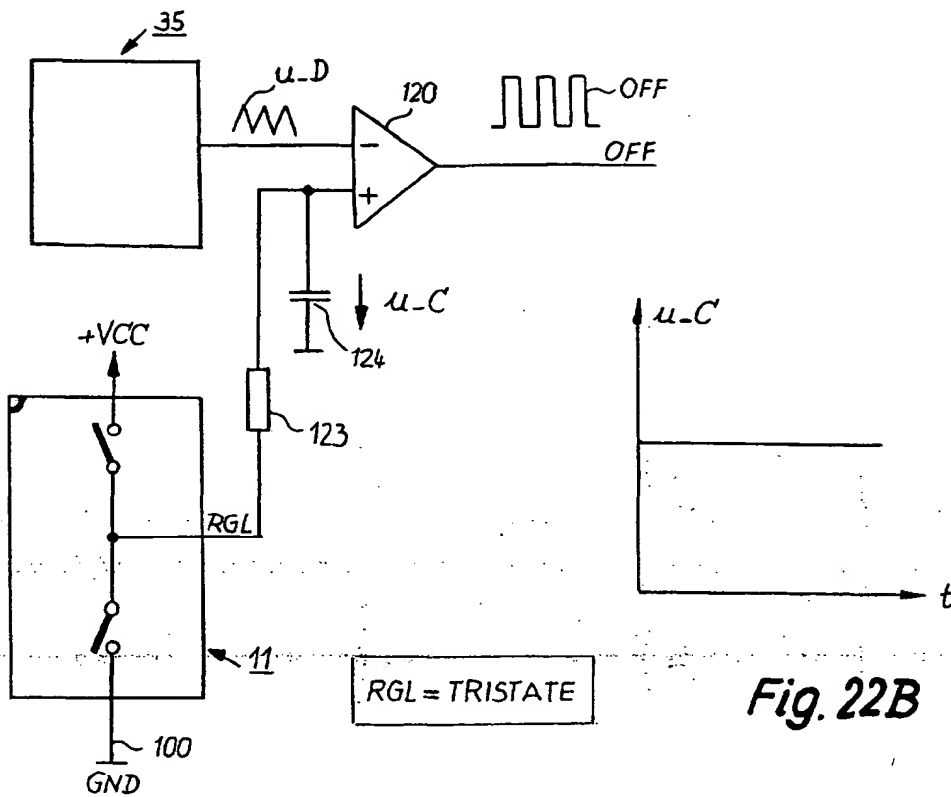


Fig. 22A

Fig. 22B

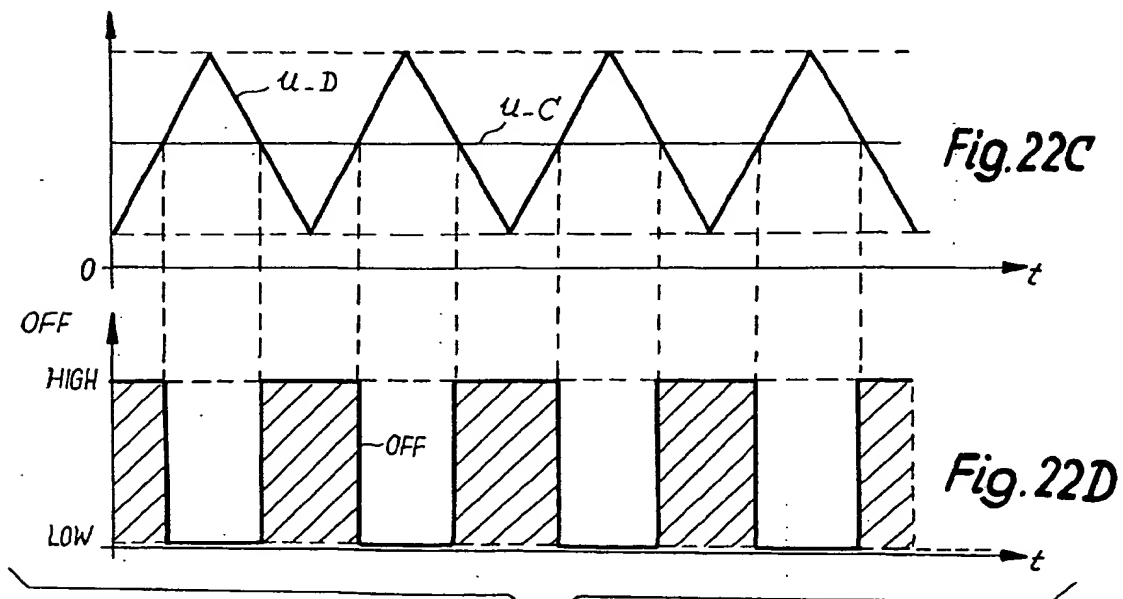


Fig. 22

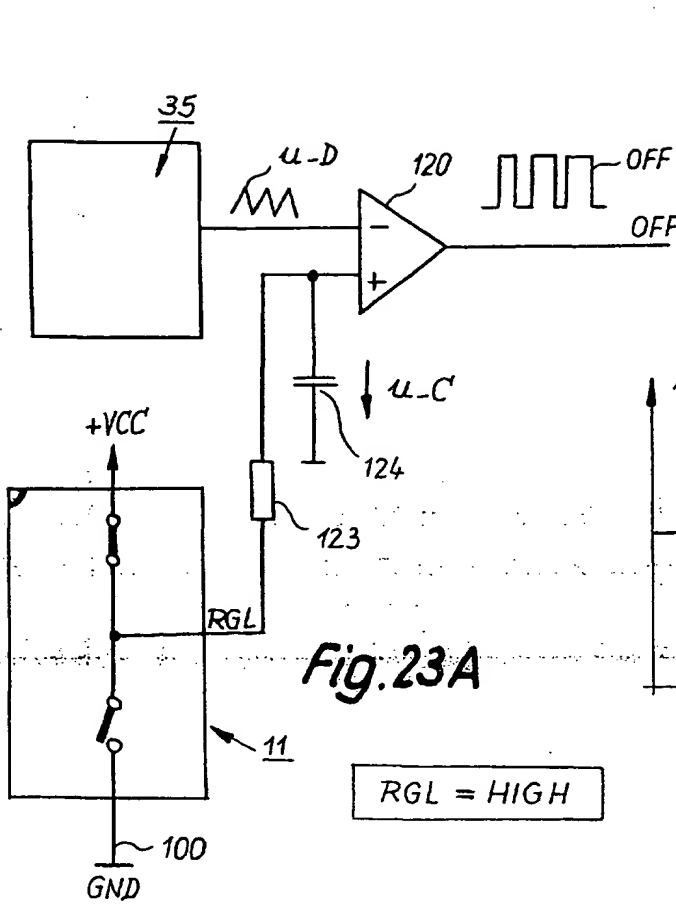


Fig. 23A

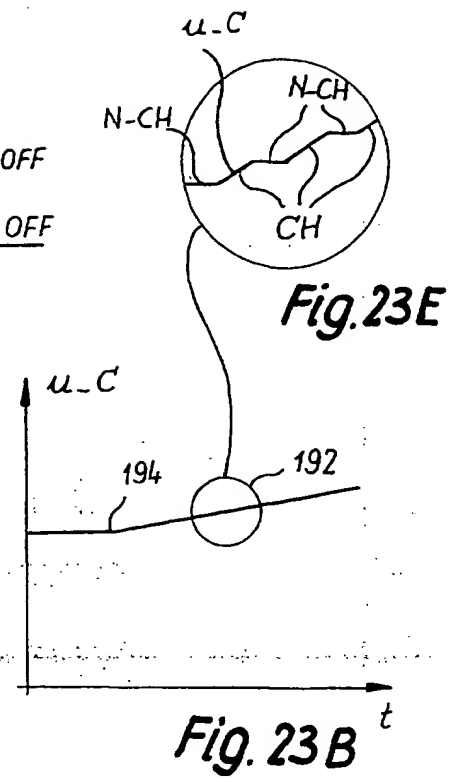


Fig. 23E

Fig. 23B

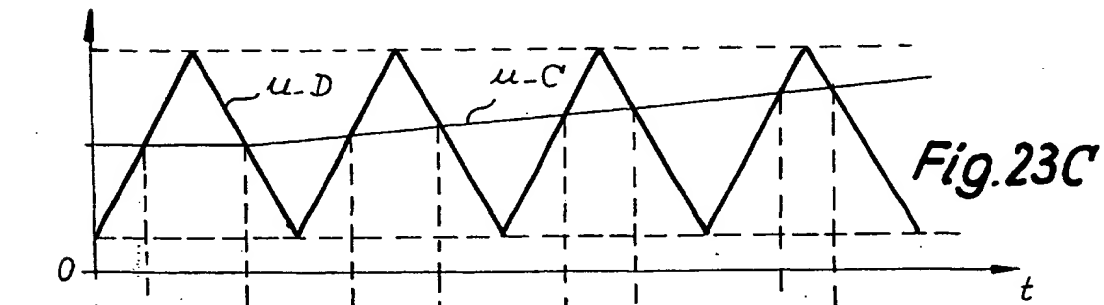


Fig. 23C

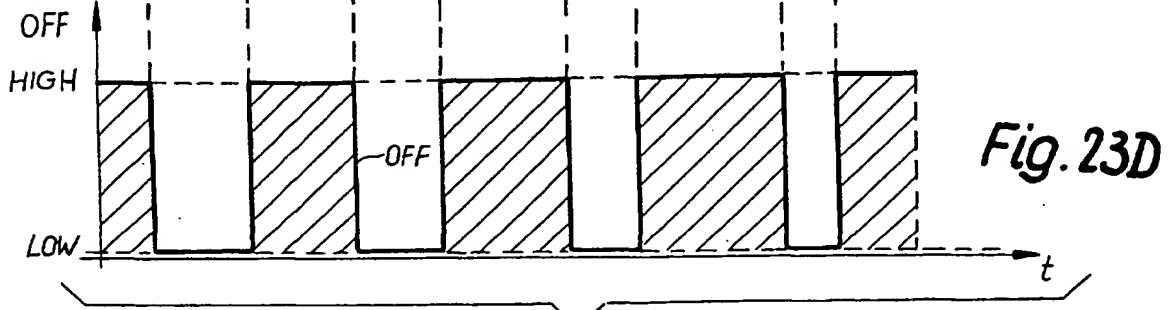
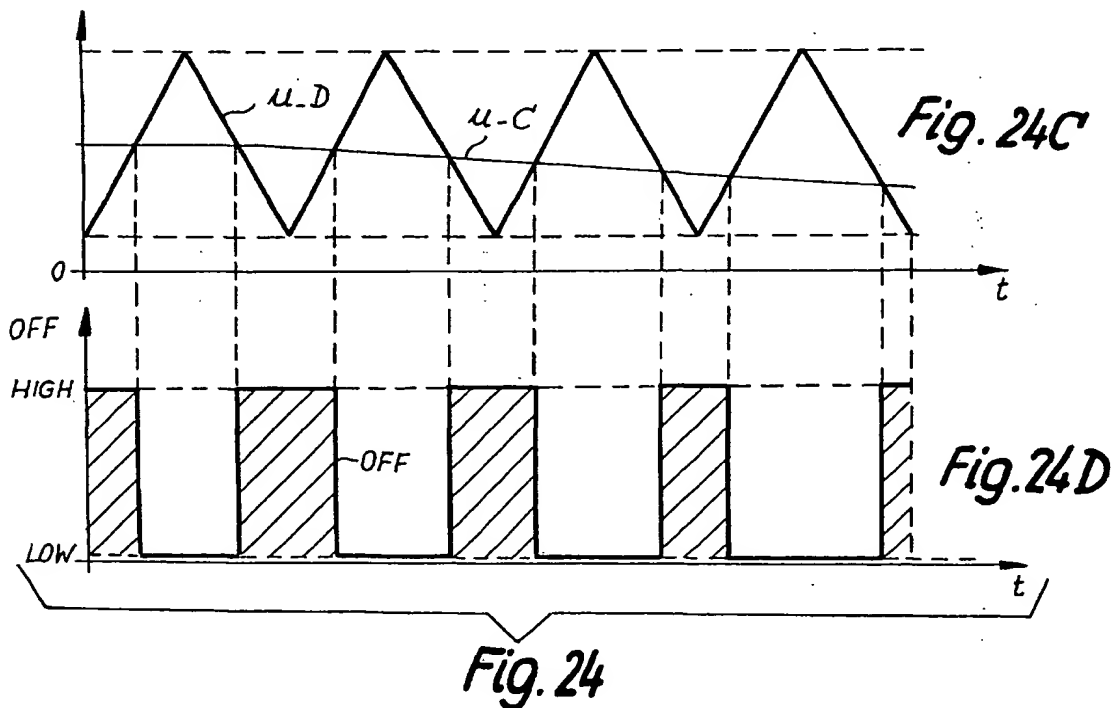
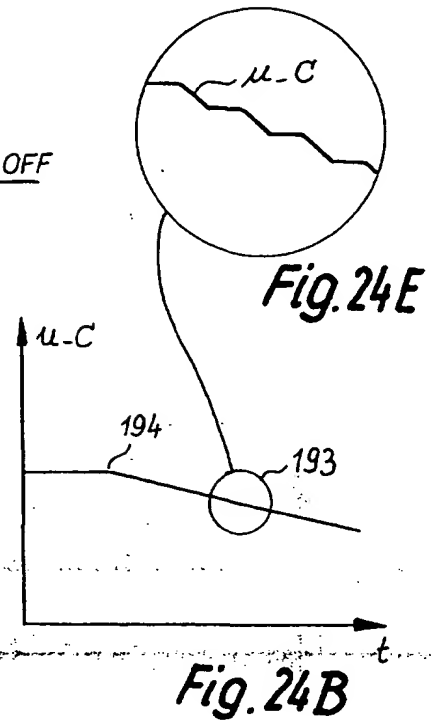
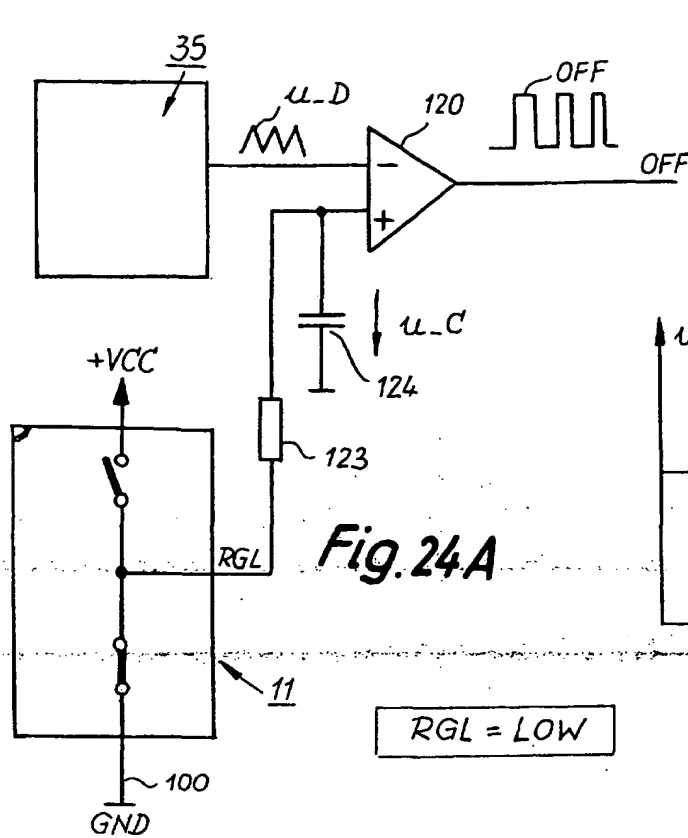


Fig. 23D

Fig. 23



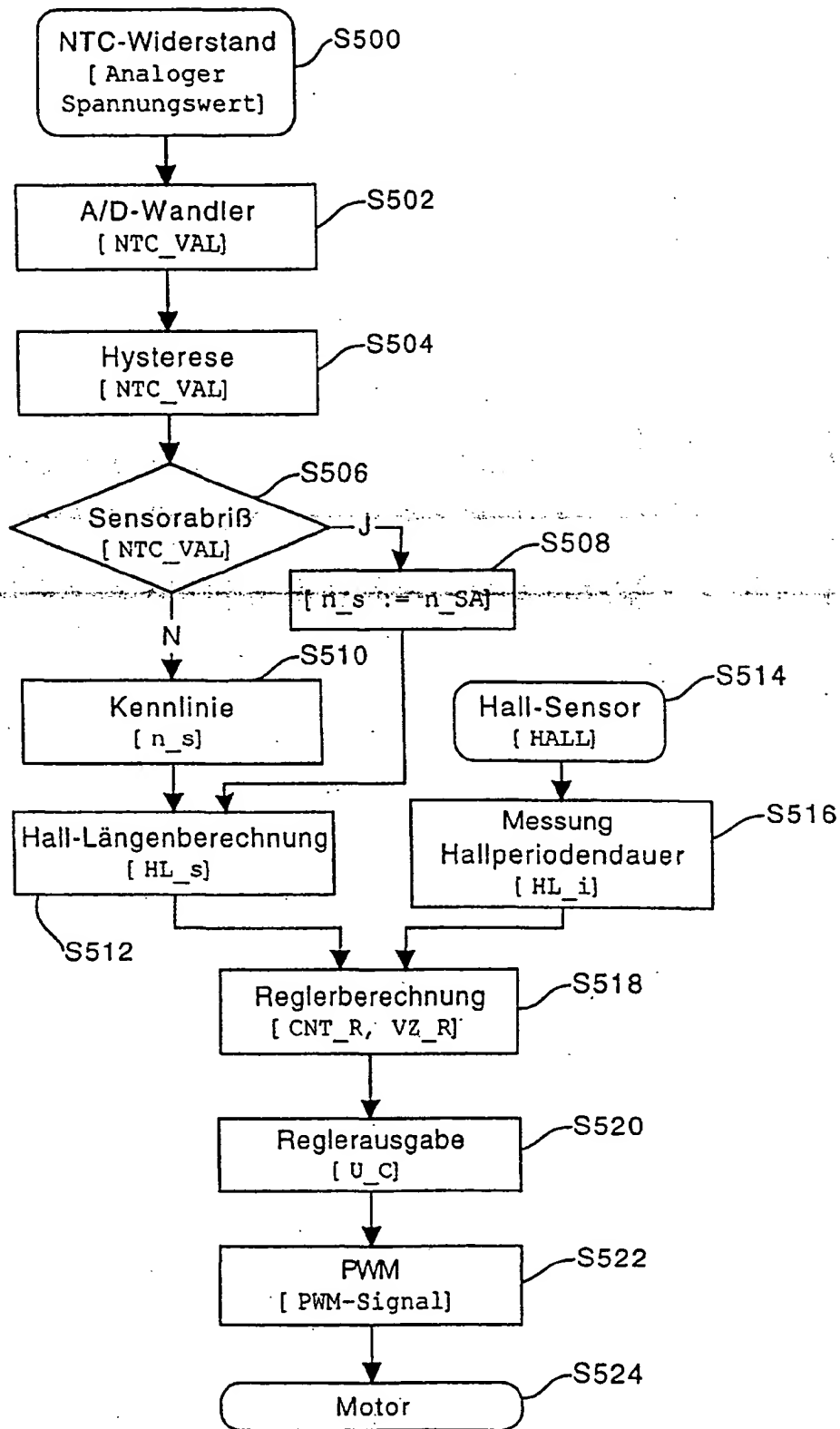


Fig. 25

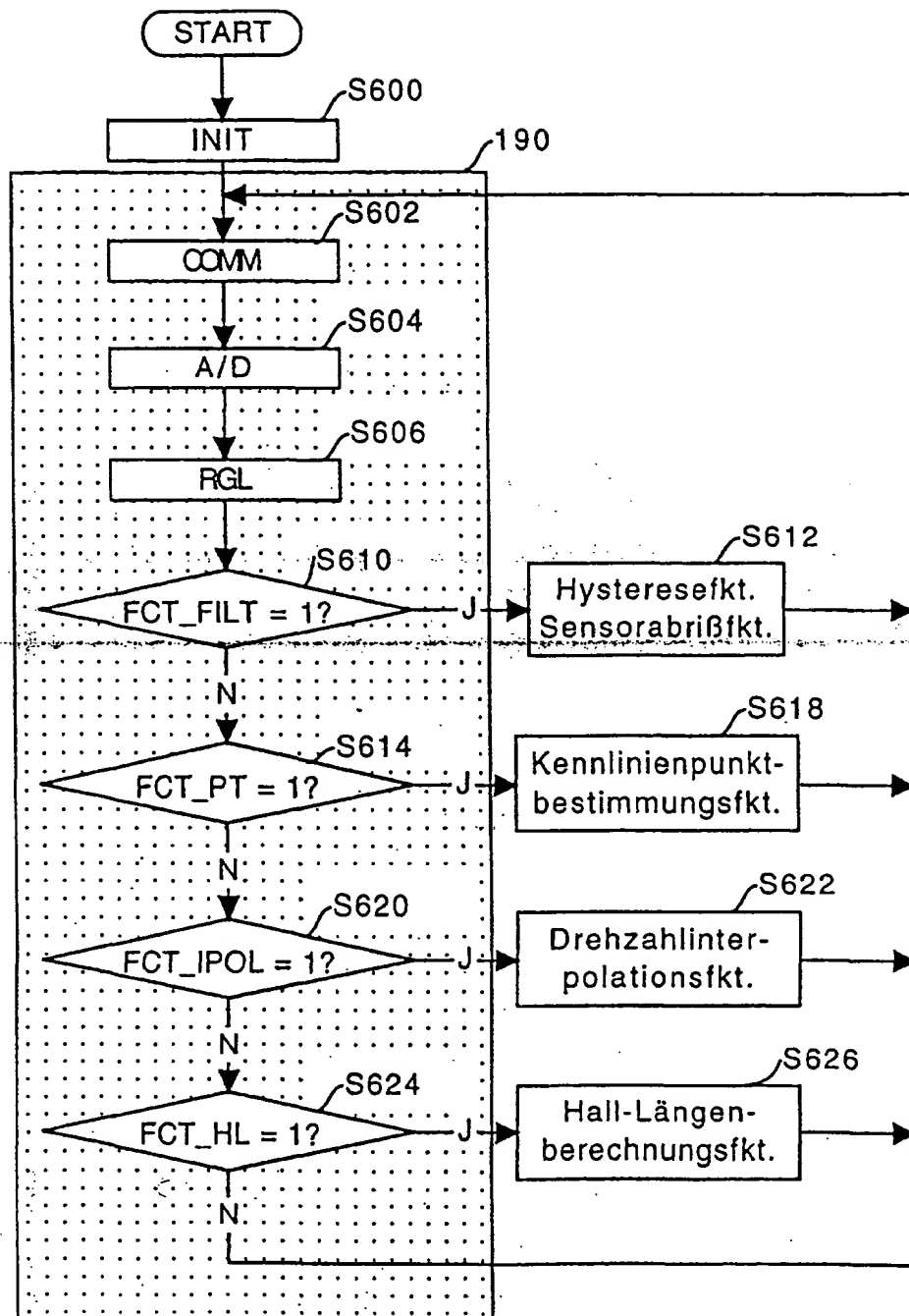


Fig. 26

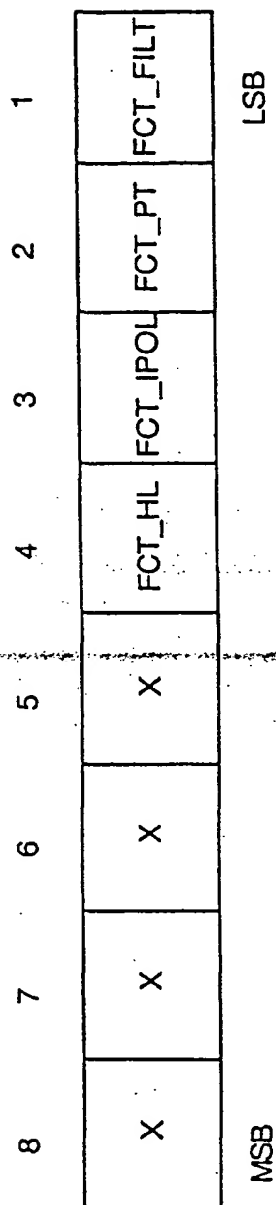


Fig. 27

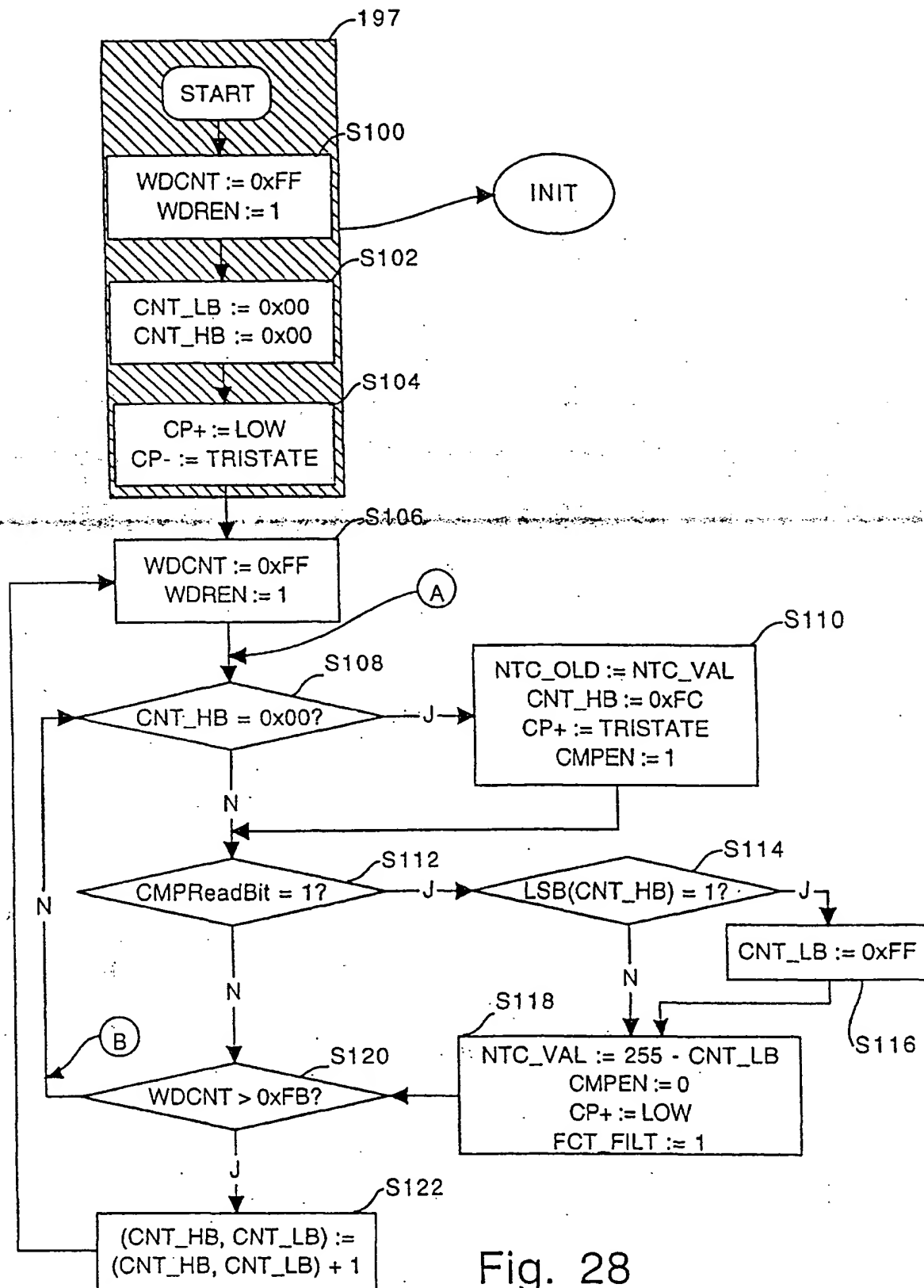


Fig. 28

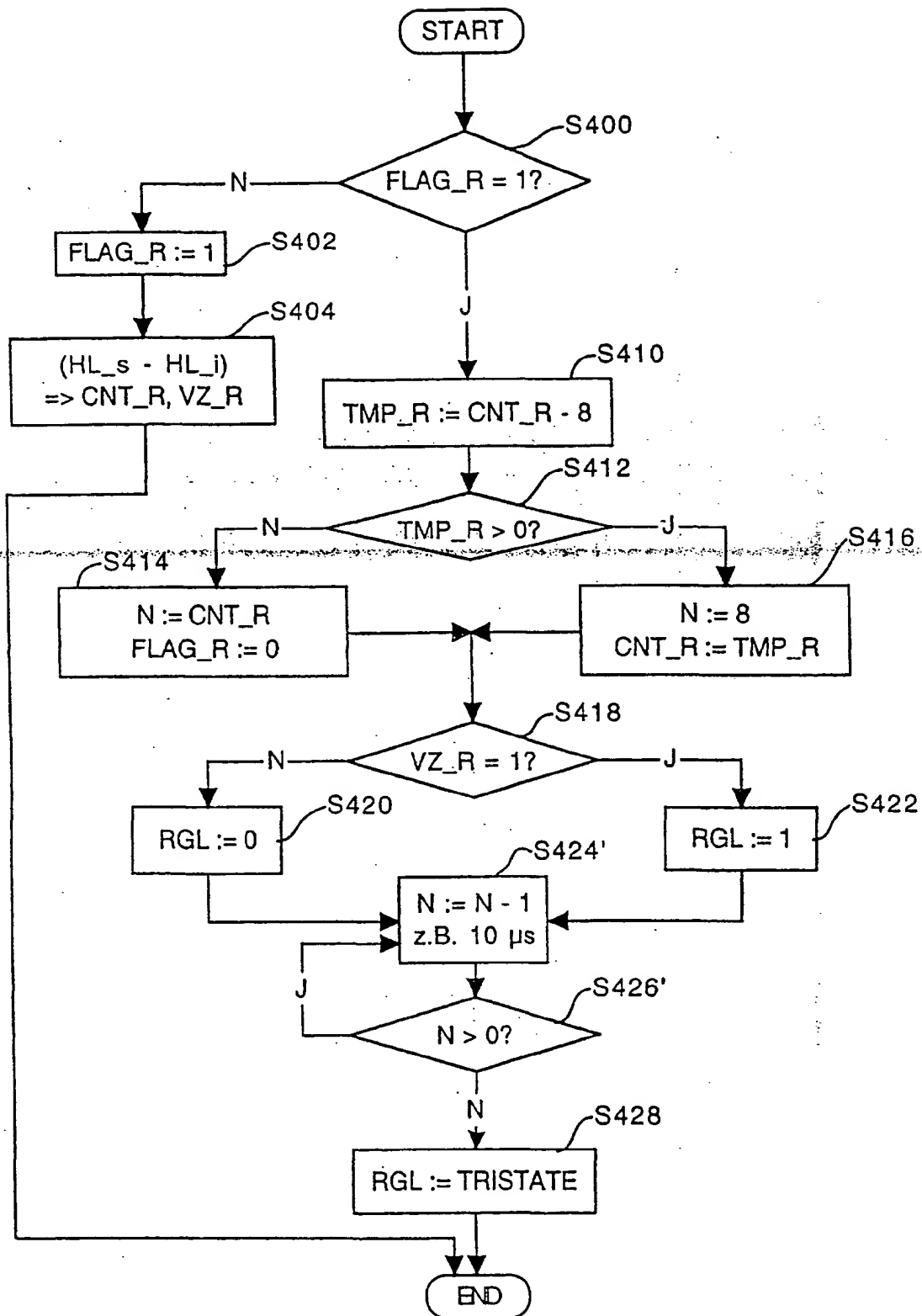


Fig. 29